

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Smartphone como soporte para Interacción Afectiva: Captura
de señales con interés emocional**

**Álvaro Rosa Fernández
Tutor: Jaime Moreno Llorena**

Enero 2015

Smartphone como soporte para Interacción Afectiva: Captura de señales con interés emocional

Álvaro Rosa Fernández
Tutor: Jaime Moreno Llorena

RESUMEN

Vivimos en un mundo en el que los Smartphone han cobrado un papel muy importante en nuestra vida diaria. Las funciones que ofrecen y la facilidad de uso han dado pie a que prácticamente sea indispensable para la mayoría de la población.

Gracias a su popularización, se ha conseguido que las personas lleven voluntariamente y en todo momento un dispositivo provisto de sensores de todo tipo y en continuo crecimiento.

El área de investigación de la Interacción Afectiva puede aprovechar los datos que ofrecen dichos sensores que de otra forma no podría obtener debido a lo complicado, costoso y molesto que sería proveer de sensores externos a los usuarios.

El objetivo de este trabajo es el análisis de las capacidades de los Smartphone como dispositivos de captura de señales con interés emocional y la creación de una librería para el reconocimiento de estados de ánimo deducidos a partir de los datos recogidos por los sensores. Dicha librería será diseñada de tal forma que pueda ser extensible para dar cabida a los nuevos sensores que incorporen los Smartphone del futuro y de esta manera poder recoger más señales con interés emocional. Asimismo, el motor de inferencia también estará diseñado para poder ser modificado y reinterpretar las señales emocionales para determinar el estado emocional del usuario.

Como prueba de las funcionalidades provistas por la biblioteca, se creará la aplicación para Android “AppMood”, la cual trabajará como un servicio transparente al usuario, recogiendo datos de los sensores e interpretando el estado de ánimo del usuario en todo momento.

Palabras clave: Smartphone, emoción, sensor, computación afectiva, vestible.

ABSTRACT

We are living in a world where the Smartphones have become a very important role in our everyday lives. The performances that they provide and the ease of use have caused that they are essential for most people.

Thanks to its popularization, the people carry a device that have all kind of sensors, all the time and voluntarily.

The Affective Interaction area can take advantage of the data provided by this sensors that otherwise could not obtain due to the complicated, expensive and annoying that could be the provision of extern sensors to the users.

The objective of this paper is the analysis of the capabilities of the Smartphones like a device for the collection of signals with emotional interest and the creation of a library for mood recognition derived from the data provided by the sensors. This library will be designed extensible for the addition of the new sensors that will provide the future Smartphones, in order to obtain more signals with emotional interest. Also, the inference engine shall be modified for reinterpret the user's mood by the emotional signals.

For test the functions of the library, an Android application (AppMood) will be developed. This application will run like a background service, gathering data from sensors and calculating the user's mood every time.

Keywords: Smartphone, emotion, sensor, affective computing, wearable.

AGRADECIMIENTOS

En primer lugar quiero agradecer a mi familia, especialmente a mis padres, el apoyo y la confianza depositados en mí a lo largo de estos años y el haberme proporcionado estos estudios.

Por otra parte me gustaría agradecer a la gran parte del equipo docente de la Escuela Politécnica Superior de la UAM su esfuerzo y dedicación para intentar que sus alumnos aprendiéramos lo mejor y lo máximo posible. Me gustaría destacar especialmente a mi tutor en este TFG, Jaime Moreno, por todos sus consejos, ayudas y comentarios brindados a lo largo de la realización de este trabajo.

Por último, cómo olvidar a toda esas maravillosas *Criaturas de Mordor* con la que tantos “sufrimientos” he compartido: Antonio, Pablo, Alberto, Miguel, Jaime, Ainoa, Carlos, JC, Koke, Luis, Javi y alguno más que me dejaré, de los que surgieron proyectos interesantes como *Las Primas de Cazorlita* o *Welele*.

Índice

RESUMEN.....	5
ABSTRACT	6
CAPÍTULO 1	13
1.1. Motivación	13
1.2. Objetivos	14
1.3. Estructura de la memoria.....	15
CAPÍTULO 2	16
2.1. Computación afectiva.....	16
2.2. Reconocimiento facial.....	18
2.3. Reconocimiento por voz	19
2.4. Vestibles	21
2.5. Campos magnéticos.....	21
2.6. Android	22
2.7. Clasificadores de emociones	22
CAPÍTULO 3	24
3.1. Análisis	24
3.1.1 Análisis de los Smartphone	25
3.1.2 Análisis de los sensores de Android y señales humanas	26
3.1.3 Análisis del motor de inferencia	30
3.1.4 Análisis de requisitos.....	33
3.1.5 Análisis de tecnologías	36
3.2. Diseño	38
3.2.1 Diseño de la biblioteca	39
3.2.2 Diseño de la aplicación AppMood	42
3.3. Implementación	46
3.3.1 Implementación de la biblioteca	46
3.3.2. Implementación de AppMood	58
3.3.3 Otras consideraciones sobre la implementación (interfaces y recursos).....	60
CAPÍTULO 4	62
4.1. Pruebas unitarias	62
4.2. Pruebas de la aplicación AppMood	68
CAPÍTULO 5	71
5.1. Conclusiones.....	71
5.2. Trabajo futuro	71
REFERENCIAS	73

ÍNDICE DE FIGURAS

FIGURA 2.1: SOFTWARE DE AFFECTIVA EN ACCIÓN. FUENTE: [12].....	17
FIGURA 2.2: VESTIBLE Q SENSOR. FUENTE: [12].....	17
FIGURA 2.3: VESTIBLE E4 WRISTBAND. FUENTE: [13].....	18
FIGURA 2.4: RUEDA DE LAS EMOCIONES DE PLUTCHIK. FUENTE: [20].....	23
FIGURA 2.5: CUBO DE EMOCIONES DE LÖVHEIM. FUENTE: [20].....	23
FIGURA 3.1: MODELO CIRCUMPLEJO DE RUSSELL BASADO EN DOS EJES. FUENTE: [6].....	31
FIGURA 3.2: MODELO DE SOFTWARE AFECTIVO. FUENTE: PROPIA.....	38
FIGURA 3.3. EMOCIONES DISTRIBUIDAS EN EL MODELO DE RUSSELL. FUENTE: [6] Y PROPIA...	40
FIGURA 3.4. MAQUETA DE LA PRIMERA VERSIÓN DE LA APLICACIÓN.....	41
FIGURA 3.5: CASOS DE USOS DE LA APLICACIÓN APPMOOD.....	43
FIGURA 3.6: DIAGRAMA DE CLASES DE LA APLICACIÓN APPMOOD.....	44
FIGURA 3.7: MAQUETA DE LAS FUNCIONALIDADES AÑADIDAS POR LA APLICACIÓN APPMOOD.....	45
FIGURA 3.8. ACELERÓMETRO EN FUNCIONAMIENTO.....	48
FIGURA 3.9. SENSOR DE LUMINOSIDAD EN FUNCIONAMIENTO.....	49
FIGURA 3.10. SENSOR DE PROXIMIDAD EN FUNCIONAMIENTO.....	50
FIGURA 3.11. SENSOR DE CAMPOS MAGNÉTICOS EN FUNCIONAMIENTO.....	50
FIGURA 3.12. RECONOCEDOR FACIAL EN FUNCIONAMIENTO.....	52
FIGURA 3.13. BÚSQUEDA POR VOZ EN FUNCIONAMIENTO.....	54
FIGURA 3.14. MEDIDOR DE DECIBELIOS EN FUNCIONAMIENTO.....	55
FIGURA 3.15. RESULTADOS GUARDADOS EN LA BASE DE DATOS.....	56
FIGURA 3.16. RESULTADO DE EJECUTAR LA VISUALIZACIÓN DE RESULTADOS DEL MOTOR DE INFERENCIA CON DATOS DE EJEMPLO.....	57
FIGURA 3.17. APARIENCIA DEL MENÚ LANZADO POR MAINACTIVITY.JAVA.....	59
FIGURA 3.16. MENSAJES Y NOTIFICACIONES LANZADAS POR SERVICIOSSENSOR.JAVA.....	60
FIGURA 4.1. RESULTADOS DE LA PRIMERA PRUEBA DEL ACELERÓMETRO.....	62
FIGURA 4.2. RESULTADOS DE LA SEGUNDA PRUEBA DEL ACELERÓMETRO.....	63
FIGURA 4.3. RESULTADOS DE LA PRIMERA PRUEBA DE LA CÁMARA.....	64
FIGURA 4.4. RESULTADOS DE LA SEGUNDA PRUEBA DE LA CÁMARA.....	64
FIGURA 4.5. RESULTADOS DE LA PRUEBA AL SENSOR DE PROXIMIDAD.....	65
FIGURA 4.6. RESULTADOS DE LA PRUEBA AL MICRÓFONO.....	65

FIGURA 4.7. RESULTADOS DE LA PRUEBA AL SENSOR DE CAMPOS MAGNÉTICOS.....	66
FIGURA 4.8. RESULTADOS DE LA PRUEBA AL SENSOR DE LUMINOSIDAD.....	67
FIGURA 4.9. RESULTADOS DE LA PRUEBA AL MEDIDOR DE DECIBELIOS.....	68
FIGURA 4.10. RESULTADOS DE LA PRUEBA A LA BASE DE DATOS.....	68
FIGURA 4.11. RESULTADOS DE LA PRUEBA INTERPRETANDO UNA EMOCIÓN.....	69
FIGURA 4.12. RESULTADOS DE LA PRUEBA DEL SENSOR INTERPRETANDO UNA EMOCIÓN.....	70

ÍNDICE DE TABLAS

TABLA 2.1: CARACTERÍSTICAS DE LA VOZ EN VARIAS EMOCIONES. FUENTE: [4].....	20
TABLA 3.1: COMPARATIVA DE LAS CARACTERÍSTICAS CON INTERÉS PARA EL DESARROLLO EN CADA S.O. FUENTE: PROPIA.....	25
TABLA 3.2: LISTA DE SENSORES PROVISTOS POR ANDROID Y SU UTILIDAD. FUENTE: [14] Y PROPIA.....	26
TABLA 3.3. SENSORES OBTENIDOS MEDIANTE LA APP SENSOR BOX DE LOS PRINCIPALES SMARTPHONE DEL MERCADO. FUENTE: PROPIA.....	27
TABLA 3.4: DATOS QUE PUEDEN OFRECEN LOS SENSORES A LOS EJES DEL MODELO DE RUSSELL. FUENTE: PROPIA.....	33

CAPÍTULO 1

INTRODUCCIÓN

1.1. Motivación

Las emociones influyen en todos los aspectos de nuestra vida, como en nuestra percepción, aprendizaje, funciones cognitivas y determinan la mayoría de decisiones que acabamos tomando. Gracias a ello y a las capacidades de la tecnología actual para capturar datos que determinen el estado de ánimo de cada persona en cada momento, se han convertido en uno de los canales más importantes para obtener información de los individuos.

El principal problema de esta tecnología radicaba en la dificultad de proveer de sensores a los usuarios para la captura de datos. Pero este problema se ha reducido en parte gracias a la aparición y popularización de los dispositivos inteligentes (tanto teléfonos como tabletas) que todos llevamos encima la mayor parte del tiempo, provistos de un gran número de sensores. Tanto es así, que en España la penetración de los Smartphone ha alcanzado el 84%, duplicándose en apenas 2 años, de 2011 a 2013 en el que era de solo el 41% [11].

Varios estudios indican que la inteligencia requiere de las emociones para la resolución creativa y flexible de problemas. El área de la Computación Afectiva propone que para crear máquinas genuinamente inteligentes, que pueden adaptarse e interactuar naturalmente con nosotros, se deben parecer lo máximo posible a los humanos. Para ello deben ser capaces de reconocer, expresar y “tener” emociones, lo que se conoce como Inteligencia Emocional [1]. Este trabajo se centra en la primera de esas capacidades, el reconocimiento.

Las aplicaciones y programas que se desarrollan comúnmente no suelen aprovechar el potencial de la inteligencia emocional para dotarse de la capacidad de obtener información importante del usuario para ofrecerle una mejor experiencia de uso, como podría ser por ejemplo adaptar el contenido que visualiza un usuario sabiendo cómo se siente al ver determinado contenido o recomendarle ciertos elementos si el sistema ha inferido que otros elementos similares le han gustado, donde estos elementos pueden ser vídeos que el usuario haya visto o canciones que haya escuchado, por ejemplo.

A la vista de la popularidad y el potencial de los Smartphone, se puede concluir que la adición de funciones de carácter emocional (de reconocimiento, en el caso de este trabajo) a nuestros dispositivos móviles llevará a la creación de máquinas más potentes y más parecidas a nosotros, que puedan percibir o interpretar nuestras emociones a partir de los datos que obtienen desde sus sensores y responder de forma adecuada a ellas. Por ello, la creación de una biblioteca de inferencia del estado emocional del usuario, que pueda ser usada por las aplicaciones desarrolladas para los Smartphone, dotará a estas aplicaciones de la capacidad de adaptarse al usuario y ofrecerle así una mejor experiencia de uso.

1.2. Objetivos

El objetivo perseguido por este trabajo es la demostración de las capacidades de los dispositivos móviles como máquinas sensibles a las emociones humanas, en concreto, para inferir las emociones de los usuarios.

Para ello se hará un estudio de los sensores más comunes de los Smartphone y cómo se puede capturar con ellos señales de interés emocional de las personas, para luego extrapolar esos datos a un clasificador de emociones e intentar determinar así el posible estado de ánimo del usuario.

Con ello se pretende ofrecer soluciones y mejoras para aplicaciones que utilizamos en nuestro día a día, desde una perspectiva tecnológica poco explotada.

El sistema desarrollado será una biblioteca y está pensado para ser ampliable fácilmente con nuevos sensores que aparezcan en el futuro en los dispositivos móviles o con sensores externos como pueden ser los de la plataforma Arduino, además de poder modificarse su motor de inferencia emocional o añadir nuevos motores que determinen las emociones de distintas maneras.

Para la demostración de la inferencia de emociones a partir de los datos ofrecidos por los sensores de los dispositivos móviles, se desarrollará una pequeña aplicación configurable de recomendación y reproducción de música para el sistema operativo Android que hará uso de los sensores más comunes para determinar el estado de ánimo del usuario cada cierto tiempo y actuar de la forma más adecuada con él.

1.3. Estructura de la memoria

Este documento estará dividido en los siguientes capítulos:

- **Introducción:** capítulo en el que se comentan la motivación y objetivos del trabajo.
- **Estado del arte:** en este capítulo se muestra el estado actual de los estudios y aplicaciones relacionados con computación afectiva.
- **Desarrollo de la propuesta:** en este capítulo se comenta el análisis, diseño, prototipos e implementación del sistema.
- **Pruebas:** en este capítulo se comentan las pruebas realizadas a las aplicaciones generadas y si sus respuestas han sido las esperadas.
- **Conclusiones y trabajo futuro:** en este capítulo se comentan las conclusiones del trabajo realizado, así como los posibles usos y mejoras del software en el futuro.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se muestra el estado de los estudios y aplicaciones actuales que hacen uso de las señales con interés afectivo que se pueden captar de los usuarios. En primer lugar se mostrarán usos que se están dando actualmente a la computación afectiva, después se comentarán las formas más utilizadas para la captura de señales y por último se hablará de trabajos actuales desarrollados para el sistema operativo Android, así como de los clasificadores de emociones más importantes desarrollados hasta la fecha. Mediante este estudio se pretende conseguir una idea para el desarrollo de la aplicación propuesta y observar las principales problemáticas con el fin de intentar exponer soluciones.

2.1. Computación afectiva

El área de la computación afectiva está formada por una línea de estudio que engloba la informática, psicología y la ciencia cognitiva. Aunque es un área estudiada desde los años 90 por el Instituto Tecnológico de Massachusetts [1], no ha sido demasiado explotada comercialmente, reduciéndose prácticamente a casos de investigación. En 2009, la profesora Rosalind W. Picard, del propio MIT, junto con la investigadora científica Rana El Kaliouby fueron de las primeras personas en fundar una compañía dedicada exclusivamente a la computación afectiva (*Afectiva*), comercializando con éxito lo que denominan “tecnologías de la emoción”.

Basándose en que la cara es una ventana a nuestras emociones, en *Afectiva* han desarrollado una aplicación de reconocimiento facial que determina en cualquier momento, mediante cualquier interfaz (puesto que funciona en ordenadores, tabletas, móviles...) la emoción que le produce al usuario el contenido que está visualizando [12].

La aplicación se puede probar on-line en la propia página de la compañía y colaborar con los datos extraídos de nuestras emociones. Para ello, mientras se están visualizando vídeos, el sistema accede a la webcam para medir las sonrisas y su intensidad, así como los ceños fruncidos y las arrugas que se marquen en la cara y determinar los niveles de sorpresa, diversión o confusión. Estos datos después serán

utilizados para recomendar campañas de publicidad para ciertas zonas de población o para crear mejores tráileres con las partes que más han gustado a los usuarios de los que han recogido los datos, por ejemplo.



Figura 2.1: software de *Affectiva* en acción. Fuente: [12]

Para mejorar el análisis de emociones, el equipo de *Affectiva* también ha creado un vestible (o *wearable*) llamado “*Q sensor*”.



Figura 2.2: vestible *Q sensor*. Fuente: [12]

Este brazalete, que se lleva como un reloj de pulsera, monitoriza los cambios en el estado anímico mediante la actividad electrodérmica (EDA), basándose en las señales eléctricas que envía el cerebro a la piel. La EDA se puede medir de muchas maneras diferentes, incluyendo el potencial de la piel, la resistencia, la conductancia, la admisión, y la impedancia. El “*Q sensor*” proporciona una forma de capturar la conductancia eléctrica mediante el paso de una cantidad minúscula de corriente entre los dos electrodos en contacto con la piel.

Motivada por la creciente demanda de usos para el “*Q sensor*”, Rosalind W. Picard fundó en 2013 otra compañía destinada a la computación afectiva, *Empatica*. Esta empresa está centrada en la creación de nuevos vestibles para una mejor recogida de datos de carácter afectivo y fisiológico y su uso en el área de la salud [13].

Mediante la creación del E4 Wristband, pretenden recoger muchos más datos interesantes que con el “*Q sensor*”. Este brazalete recoge continuamente el ritmo cardíaco, la actividad electrodérmica, la temperatura y la actividad física mediante tres acelerómetros. Con las aplicaciones desarrolladas para este sensor, se puede monitorizar la salud del paciente mientras está en su casa, o mediante una aplicación móvil por los familiares.



Figura 2.3: vestible *E4 Wristband*. Fuente: [13]

Los proyectos realizados por los clientes de *Empatica* van desde la monitorización del estrés de los usuarios en su vida diaria hasta la detección temprana del párkinson, pasando por ayudas en la comunicación de las personas autistas.

2.2. Reconocimiento facial

El reconocimiento de emociones a partir de la cara de una persona sería la primera funcionalidad que se debería añadir a un sistema de inferencia emocional, dado que, como dicen los creadores de *Affective*, la cara es la ventana a nuestras emociones y nosotros como humanos, nos fijamos principalmente en la cara de los demás para interpretar su estado de ánimo. Además, la cara siempre suele estar visible y omnipresente, mientras que por ejemplo, la voz, es intermitente. Mediante los músculos de la cara, se pueden formar más de mil gestos faciales y por la acción tan rápida de estos músculos, podrían mostrarse todos estos gestos en pocas horas [2] [3].

La principal problemática que supone el uso del reconocimiento facial es que se debe disponer de una cámara que esté grabando continuamente nuestras caras. Por lo que los usos de esta tecnología se reducen a los que se puedan desarrollar para estar delante de la webcam del ordenador o de la cámara del móvil.

La creación de un vestible como son las *Google Glasses*, ha permitido una mayor zona de desarrollo para esta tecnología.

Aparte de los usos ya mencionados desarrollados por *Affective*, el Instituto de Sistemas Integrados Fraunhofer está trabajando en un software (*SHORE*) integrado en las Google Glasses, capaz de leer el rostro humano, para valorar mejor a nuestros interlocutores y saber qué están sintiendo. Para ello, el sistema se ha entrenado con 10.000 caras reales.

Primeramente, el software *SHORE* detecta la edad y sexo de la persona o personas que están en el radio de visión de las gafas. Después, mediante el análisis de los gestos de la cara, asigna porcentajes a los valores de enfado, felicidad, tristeza y sorpresa. También tiene la funcionalidad de detectar reacciones muy pequeñas para tratar de saber si la persona está ocultando algo o mintiendo [23].

Cabe destacar que nVISO es otra compañía dedicada al uso del reconocimiento de emociones mediante los cambios musculares de la cara, al igual que *Affective* para usos publicitarios [24].

Un gran aporte al reconocimiento emocional mediante el reconocimiento facial ha sido la creación de la biblioteca libre y gratuita *OpenCV*. Se trata de una biblioteca multiplataforma creada por *Intel* en 1999, compuesta por más de 500 funciones que permiten visión artificial y que abarcan áreas como la visión robótica o reconocimiento de objetos [22]. Gracias a la implementación para su uso en el entorno Android, y a las funciones de reconocimiento de partes de la cara, se ha usado en este proyecto. Su implementación se puede observar en el apartado *desarrollo*.

2.3. Reconocimiento por voz

Cuando hablamos es importante “qué” decimos, pero también “cómo” lo decimos, ese es un buen canal por el que se pueden captar nuestras emociones [4].

The effect of emotions on the human voice

	fear	anger	sorrow	joy	disgust	surprise
speech rate	much faster	slightly faster	slightly slower	faster or slower	very much slower	much faster
pitch average	very much higher	very much higher	slightly lower	much higher	very much lower	much higher
pitch range	much wider	much wider	slightly narrower	much wider	slightly wider	
intensity	normal	higher	lower	higher	lower	higher
voice quality	irregular voicing	breathy chest tone	resonant	breathy blaring	grumbled chest tone	
pitch changes	normal	abrupt on stressed syllable	downward inflections	smooth upward inflections	wide downward terminal inflections	rising contour
articulation	precise	tense	slurring	normal	normal	

Tabla 2.1: características de la voz en varias emociones. Fuente: [4]

El software desarrollado hasta ahora se basa en la comparación de los datos recogidos por el micrófono y su comparación con las grabaciones (recogidas o interpretadas por actores) de cada una de las emociones que se quieren reconocer [5].

Para ello, se recogen datos del Pitch (frecuencia a la que vibran las cuerdas vocales, frecuencia fundamental) del usuario y se obtienen los siguientes valores: valor medio, rango y variabilidad.

- Un valor medio alto indicará excitación.
- Un rango alto indicará excitación.
- Una variabilidad alta indicará emoción negativa.

Una vez obtenidas las características de la voz, se clasifican y comparan con los modelos obtenidos anteriormente y se asigna la emoción con mayor semejanza.

Este es el sistema utilizado, por ejemplo, por *EmoSpeech*, una empresa que provee información a los *call centers* sobre si sus llamadas interesan a determinados usuarios mediante el estudio de su voz [25].

2.4. Vestibles

Un vestible o *wearable* es un tipo de ordenador/dispositivo que puede ser usado sobre el cuerpo como si se tratara de una prenda de vestir.

Debido a lo costosos e incómodos que eran de llevar los vestibles, su uso se vio reducido a unas pocas situaciones, como podían ser los entornos militares, sanitarios o para los trajes de los astronautas. En la última década, estos dispositivos se han mejorado enormemente y han sido rediseñados para ser más pequeños y ergonómicos y poder ser comercializados.

Los vestibles pueden ser utilizados para mejorar nuestras capacidades. Por ejemplo, podemos ver con visión infrarroja, o con una cámara en las gafas o en la cabeza se pueden grabar automáticamente los momentos detectados como más emocionales y “aumentar” así la memoria humana [1].

Actualmente, aparte de las pulseras ya nombradas en el apartado 2.1 usadas por las empresas *Afectiva* y *Empatica*, también podemos encontrar las Google Glasses y relojes inteligentes fabricados por la mayoría de los fabricantes de teléfonos. El interés afectivo de estos dispositivos es su capacidad para proveer de sensores que hagan electromiografías, controlen la presión sanguínea, conductancia de la piel y la respiración de los usuarios, datos que se pueden usar para determinar el estado de ánimo [1].

La importancia de los vestibles para este trabajo reside en que algunas de sus funcionalidades pueden ser encontradas directamente en los sensores de nuestros teléfonos inteligentes, o ser replicados mediante el uso de la gran cantidad de sensores que provee la plataforma Arduino, compatibles con el sistema operativo Android.

2.5. Campos magnéticos

Existen estudios que indican que el corazón humano tiene su propio sistema nervioso, su propio cerebro formado por unas 40.000 neuronas [7]. La matemática e investigadora Annie Marquier afirma que ese sistema nervioso emite un campo magnético que varía respecto a nuestro estado emocional. Emociones positivas emiten ondas armónicas y regulares, mientras que emociones negativas emiten ondas incoherentes y desordenadas.

No se han desarrollado demasiadas aplicaciones que usen este tipo de sensor para detección emocional, quizá porque están sostenidas por muy pocos estudios y la mayoría de dudosa fiabilidad. Solo pueden destacarse las aplicaciones que captan mediante el campo magnético el “aura” de una persona y lo representan sobre la imagen captada por la cámara.

2.6. *Android*

Los sensores y otras características de los dispositivos móviles han sido utilizados en la aplicación *Emotionsense*, creada por la Universidad de Cambridge, para la detección del estado de ánimo del usuario [8].

Durante el día, la aplicación genera notificaciones invitando al usuario a rellenar encuestas sobre cómo se siente. Después el sistema recoge datos del GPS, acelerómetro y micrófono y monitoriza las conversaciones y los SMS enviados por el usuario para desarrollar un patrón de hábitos; mediante la actividad detectada por el acelerómetro y mediante las llamadas, el sistema detecta si el usuario está activo socialmente por el uso continuado de su dispositivo. Durante la primera semana, la aplicación recolecta estos datos y los contrasta con los estados emocionales del usuario. Mediante dos ejes (positividad y actividad), el usuario define su estado de ánimo en las encuestas. Ocho semanas después, la aplicación provee al usuario los resultados de cómo se corresponden sus actividades cotidianas con sus estados de ánimo.

2.7. *Clasificadores de emociones*

Paul Ekman concluyó que existen 6 emociones básicas y diferenciadas entre sí: furia, asco, miedo, felicidad, tristeza y sorpresa y que existen características particulares relacionadas con cada una de ellas, por lo tanto mediante las distintas variantes de estas características se pueden obtener las 6 emociones básicas y sus emociones derivadas [9]. En 1897, el padre de la psicología moderna, Wilhelm Max Wundt propuso que las emociones pueden ser descritas por tres dimensiones: excitación, tensión y placer [10], pero después, la mayoría de modelos que se crearon fueron bidimensionales: el modelo circunplejo de Russell (basado en la valencia y la excitación en un espacio circular de dos dimensiones), el modelo vectorial (basado en vectores que apuntan a dos direcciones, representando una especie de bumerán) y el

modelo “activación positiva – activación negativa” (PANA) que representa mediante un ángulo entre 0° y 45° la negatividad o positividad de la emoción. Existen también modelos con tres dimensiones como el que utiliza el modelo de Plutchik con su rueda de las emociones, donde mediante círculos concéntricos representa emociones básicas (lejos del centro) y emociones complejas (cerca del centro) [20].

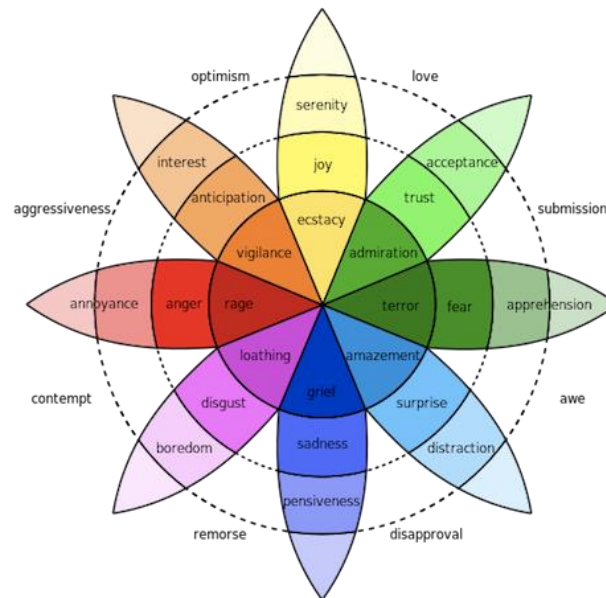


Figura 2.4: rueda de las emociones de Plutchik. Fuente: [20]

Otra forma de representar las emociones es con el cubo de Lövheim, que determina la emoción con los niveles de dopamina, noradrenalina y serotonina y con ocho emociones básicas [20].

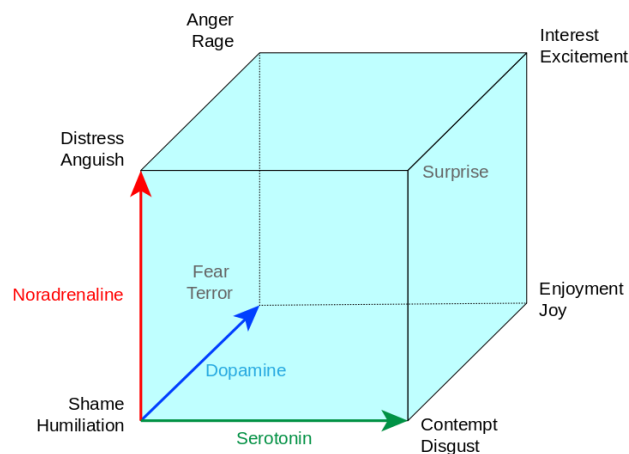


Figura 2.5: cubo de emociones de Lövheim. Fuente: [20]

CAPÍTULO 3

PROPUESTA

En este capítulo se desarrolla la propuesta de la aplicación planteada (AppMood) para alcanzar los objetivos propuestos en la introducción. El ciclo de vida de este software seguirá un desarrollo en espiral que generará primeramente una versión de la aplicación con las funcionalidades de la biblioteca: recogida de señales, almacenamiento e inferencia de emociones a partir de estos datos y en segundo lugar, generará la aplicación final que funcionará en segundo plano como un servicio recogiendo los datos solamente de los sensores que se deseen, que serán los que se haya comprobado en la primera versión como los más útiles a la hora de inferir la emoción. Esta última versión de la aplicación también deberá interactuar de alguna forma con el usuario.

3.1. Análisis

A la hora de crear una aplicación que utilice computación afectiva debemos responder antes a estas preguntas [1]:

- 1- ¿Qué emociones son relevantes para esta aplicación?
- 2- ¿Cómo pueden ser reconocidas/expresadas?
- 3- ¿Cómo responder teniendo esta información?

Para dar respuesta a estas preguntas se irán añadiendo sucesivas funcionalidades a la aplicación. Como la respuesta a la primera pregunta es “todas”, puesto que queremos reconocer todas las emociones que experimente el usuario, para avanzar en el análisis debemos responder a las otras dos preguntas. El primer prototipo tiene como objetivo responder a la segunda pregunta, ya que el objetivo principal que queremos conseguir es el reconocimiento de emociones (dejando de lado la expresión de las mismas).

Para responder a la tercera pregunta, la segunda versión de la aplicación, la versión final, deberá añadir algún tipo de utilidad que responda adecuadamente al estado emocional del usuario.

3.1.1 Análisis de los Smartphone

Dado que uno de los objetivos de este trabajo es la demostración de la utilidad de los Smartphone para inferir los estados de ánimo de los usuarios gracias a sus capacidades para recoger información interesante a partir de sus sensores, lo primero que hay que analizar es qué nos ofrecen los dispositivos más extendidos y comunes y cómo podemos utilizarlo para captar señales con interés emocional.

En primer lugar, habrá que elegir el sistema operativo más adecuado para el cual será desarrollada la aplicación. Los sistemas operativos de los Smartphone actuales son básicamente tres: Android (81,9%), iOS (12,1%) y Windows Phone (3,6%) [15]. A continuación se muestra una tabla comparando las características interesantes para el desarrollo:

S.O.	Extensión en la población	Experiencia previa en desarrollo	Software libre	Total
Android	3	1	1	5
iOS	2	0	0	2
Windows Phone	1	0	0	1

Tabla 3.1: comparativa de las características con interés para el desarrollo en cada S.O.

Fuente: propia.

Android ha recibido la más alta puntuación sumando estas tres características, por lo que será la plataforma más adecuada para el desarrollo de la aplicación: es el sistema más extendido en el mercado, por lo que la futura aplicación podrá ser utilizada por más usuarios; ya se han realizado proyectos para esta plataforma en la universidad, así que el desarrollo será más rápido y eficiente que empezando de cero en cualquiera de los otros dos sistemas; por último, Android es el único de los tres sistemas que no es software propietario por lo que su desarrollo será más sencillo al poder acceder a las bibliotecas desarrolladas y su documentación, además de poder utilizar más bibliotecas creadas por otros desarrolladores.

3.1.2 Análisis de los sensores de Android y señales humanas

Una vez elegidos los Smartphone Android como destino para la aplicación que se desarrollará, el siguiente paso sería analizar las capacidades que ofrecen para poder capturar las señales con interés emocional.

Para comprobar los sensores y utilidades que se podrán utilizar en el desarrollo de la aplicación se ha procedido a realizar un pequeño estudio sobre los sensores más comunes que tienen los terminales más extendidos en la población.

En primer lugar se ha obtenido de la propia documentación de Android la lista de sensores que pueden soportar los terminales de dicho sistema operativo y la utilidad de cada uno de ellos [14]. A la tabla se le han añadido la cámara, el micrófono y el GPS, que no son sensores propiamente dichos pero nos servirán para recoger datos con interés emocional de los usuarios, además de que prácticamente todos los terminales inteligentes del mercado los incorporan:

Sensor	¿Qué hace?
Acelerómetro	Medir aceleraciones por gravedad y cambios de movimiento
Campo magnético	Brújula, detectar campos magnéticos
Giroscopio	Detectar giros
Orientación	Indicar dirección a la que apunta el dispositivo
Luz ambiental	Ajustar iluminación pantalla respecto a la luz medida
Proximidad	Si hay un objeto a menos de 5 cm
Presión atmosférica	Altímetro, barómetro
Gravedad	Medir la aceleración debida a la gravedad
Acelerómetro lineal	Medir aceleraciones descontando la gravedad
Vector de rotación	Detectar giros
Temperatura ambiental	Medir la temperatura del aire
Humedad relativa	Medir el punto de rocío, humedad absoluta y relativa
Cámara	Captura facial
Micrófono	Captura de voz
GPS	Detectar posición

Tabla 3.2: lista de sensores provistos por Android y su utilidad. Fuente: [14] y propia.

Los sensores descritos en la tabla superior son aquellos que pueden ser incluidos en los dispositivos móviles por los fabricantes, no obstante, esto no quiere decir que todos los Smartphone dispongan de todos estos sensores. Por ello, para obtener una mejor representación de los sensores verdaderamente disponibles en los dispositivos comercializados, mediante la aplicación gratuita “*Sensor Box*” [18] se ha procedido a la comprobación de los sensores provistos por los terminales Android más extendidos en el mercado para definir cuáles son los más comunes y poder realizar así una aplicación que sea compatible con la mayor parte de los Smartphone.

Junto a Apple, Samsung es el principal fabricante dominador del mercado de los Smartphone [17], en un segundo nivel se encuentra Sony y los genéricos chinos, que en 2014 ya ocupan el 40% del mercado y en 2015 pueden acercarse al 50% [19]. Los resultados ofrecidos por la app “*Sensor Box*” en este tipo de móviles han sido los siguientes:

Sensor/Movil	NOAIN A9090	Sony Xperia Mini	Samsung Galaxy Ace 2	Huawei Ascend Y300	Samsung Galaxy S3 mini
Acelerómetro	SÍ	SÍ	SÍ	SÍ	SÍ
Luz ambiental	SÍ	NO	NO	SÍ	NO
Orientación	SÍ	SÍ	SÍ	NO	SÍ
Proximidad	SÍ	SÍ	SÍ	SÍ	SÍ
Temperatura	NO	NO	NO	NO	NO
Giroscopio	NO	NO	NO	NO	SÍ
Micrófono	SÍ	SÍ	SÍ	SÍ	SÍ
Campos magnéticos	SÍ	SÍ	SÍ	NO	SÍ
Humedad relativa	NO	NO	NO	NO	NO
Presión atmosférica	NO	NO	NO	NO	NO
Cámaras	SÍ	SÍ	SÍ	SÍ	SÍ
GPS	SÍ	SÍ	SÍ	SÍ	SÍ

Tabla 3.3. Sensores obtenidos mediante la app *Sensor Box* de los principales Smartphone del mercado. Fuente: propia

Además de todos estos sensores proporcionados por el propio dispositivo Android, hay que tener también en cuenta que esta lista se puede aumentar añadiendo los

sensores externos que pueden conectarse fácilmente a uno de estos Smartphone, como pueden ser los de la plataforma Arduino. Algunos de estos sensores que los Smartphone no proveen y que pueden recoger señales interesantes son [16]:

- Sensor de pulso cardíaco en los dedos.
- Sensor de calidad del aire.
- Sensor de temperatura.
- Sensor de humedad relativa.

Se han incluido aquí los sensores de temperatura y de humedad relativa porque aunque la tabla de sensores (3.2) refleje que los dispositivos Android los incluyen, la realidad es que prácticamente ningún dispositivo de los más extendidos en el mercado los incluye, como se puede observar en la tabla 3.3. Los otros dos sensores tampoco son provistos por los Smartphone y podrán ser útiles a la hora de determinar el estado de ánimo del usuario, por ejemplo, se podrá comparar si tiene relación la calidad del aire con el estado del usuario, y el pulso cardíaco puede determinar la excitación, el miedo o el nerviosismo.

Para poder determinar el uso que se les podrá dar a todos estos sensores a la hora de capturar señales con interés emocional de los usuarios se deberán recopilar las fuentes por las que los humanos emitimos estas señales. Estos canales podemos determinarlos utilizando los sistemas descritos en la sección *Estado del arte* y aplicando la simple lógica al preguntarnos cómo notamos los humanos el estado de ánimo de los demás. Respondiendo a esta pregunta podemos obtener las siguientes fuentes de emisión principales:

- Movimiento: ver a una persona moverse constantemente debería al menos proporcionar datos sobre su actividad. Además, el tipo de movimiento puede indicar también una emoción concreta: nerviosismo, euforia... Esto es algo utilizado por ejemplo por el vestible *Wristband* de *Empatica* con sus tres acelerómetros [13].
- Cara: como se apuntó en la sección *Estado del arte*, según *Affective* la cara es una ventana a nuestras emociones [12], o como se dice comúnmente: la cara es el espejo del alma. Si observamos la cara y los gestos faciales de alguien podemos tener una buena aproximación de cómo se siente.
- Voz: el habla de una persona también es un buen lugar del que poder deducir su estado de ánimo. Son importantes las cosas que se dicen pero también cómo se dicen [4].

- Entorno: el lugar y condiciones en las que se encuentra el usuario también pueden proporcionar valiosa información emocional. Por ejemplo, si se determina que el usuario está la mayor parte del tiempo en entornos lúgubres se podría llegar a la conclusión de que está triste, o determinar la relación de la humedad relativa con el estado emocional en ese momento.
- Señales biológicas: impulsos eléctricos (electromiografías), presión sanguínea, conductancia de la piel y la respiración son algunas de las señales que pueden ayudar a definir el estado emocional de los usuarios, algo utilizado también por el *Wristband* de *Empatica* [13].

Observando esta lista de canales por los que capturar señales de interés emocional y los datos de los sensores que incorporan los móviles más comunes de la tabla 3.3, se han obtenido las siguientes conclusiones:

- Quedan descartados para su uso en la aplicación los siguientes sensores: temperatura ambiental, humedad relativa, presión atmosférica y giroscopio. Los tres primeros por no estar presentes en ninguno de los cinco móviles reflejados en la tabla, por lo que estarán representados en muy pocos móviles del mercado y el giroscopio por estar incluido en uno solo de ellos y porque sus funciones se pueden emular mediante el acelerómetro, ya que principalmente proporcionará datos sobre movimiento.
- La cámara podrá ser utilizada en reconocimiento de emociones faciales.
- El micrófono podrá ser utilizado para reconocimiento emocional en las características de la voz (cómo se dice) y para comprobar las búsquedas realizadas por voz (qué se dice).
- Los sensores de proximidad, orientación y el acelerómetro permiten, a primera vista, monitorizar la actividad del usuario y determinar señales por movimiento.
- El sensor de luz ambiental podrá ser utilizado para comprobar la luminosidad a la que está expuesto el usuario.
- Aunque no se haya señalado como canal del que obtener información y puesto que el sensor de campos magnéticos está incluido en casi todos los dispositivos, este sensor podrá ser aprovechado para medir el campo magnético cercano al usuario e intentar medir el “aura” que le rodea, como se ha comentado en la sección del *Estado del arte* 2.5.

- Aunque el GPS está incluido en todos estos móviles, queda descartado pues su uso requiere tener activado el servicio siempre y la mayoría de los usuarios nunca lo tiene activo, principalmente por su elevado consumo de batería, por lo que solo lo activan en trayectos. Aun si se decidiera usar, solo serviría para saber si el usuario está en movimiento, algo que ya nos provee el acelerómetro, por ejemplo.
- Las señales biológicas del cuerpo requieren de hardware externo para poder ser medidas, algo que no compete a este trabajo. Aunque es algo importante de señalar puesto que la biblioteca desarrollada permitirá recibir datos de sensores que se añadan, además de que quizá en el futuro los móviles estén provistos de nuevos sensores que puedan capturar estas señales. En la sección *Trabajo futuro* se comenta en detalle este tema.

Una interfaz simple que permita probar y registrar datos interesantes de los sensores será suficiente para implementar las funcionalidades descritas hasta ahora para un primer prototipo.

3.1.3 Análisis del motor de inferencia

Para continuar desarrollando la aplicación que deseamos realizar comprobamos las etapas por las que debe pasar un software afectivo [1]:

- Entrada: canales de los que obtener las señales interesantes emocionalmente: caras, voz, gestos de la mano, postura, forma de andar, respiración, respuesta electrodérmica, temperatura, electrocardiograma, presión sanguínea...
- Reconocimiento de patrones: interpretación de las señales, como por ejemplo, diferenciar una sonrisa de otro gesto que sea parecido.
- Razonamiento: predecir emoción basándose en el conocimiento sobre cómo son generadas y expresadas. Requiere percibir y razonar sobre el contexto, y situación personal de cada uno.
- Aprendizaje: el software o dispositivo aprende de la persona y determina cuáles son los factores más importantes de los anteriores.
- Parcialidad: el “estado de ánimo” del software o dispositivo puede influenciar en el reconocimiento de emociones ambiguas.
- Salida: el software nombra o describe las expresiones reconocidas, y la emoción que corresponde.

Observando estas etapas se deberán añadir nuevas funcionalidades a la aplicación. La etapa de entrada ya la tenemos definida en el apartado anterior, habrá que introducir cambios para recoger las señales que más nos interesen. Las etapas de reconocimiento de patrones y aprendizaje se subsanarán en gran parte mediante la introducción de una base de datos que guarde todos los datos de los sensores y por lo tanto, el histórico de las emociones del usuario. La etapa de parcialidad no influye en nuestra aplicación puesto que solo se basa en un motor de inferencia y no tiene otro para la expresión, el sistema no “tendrá” emociones. La etapa de salida mostrará la emoción inferida y una respuesta adecuada por parte del dispositivo, esta etapa será acoplada en la versión final de la aplicación comentada más adelante. La etapa de razonamiento es la más importante para proseguir con la implementación y es la que se va a analizar ahora. Para definir la emoción basada en los datos recogidos por los sensores necesitamos un clasificador de emociones respecto a parámetros y cómo asignar esos parámetros. Observando la complejidad de los sistemas clasificadores de emociones descritos en la última sección del capítulo *Estado del arte*, se ha considerado que el modelo más adecuado para su uso en el desarrollo de la aplicación es el modelo circumplejo de Russell, basado en dos ejes: excitación y valencia [6], pues es la forma más sencilla de representar una emoción (por ejemplo, los niveles usados en el cubo de Lövheim son difíciles de medir, puesto que son características químicas del cuerpo, y tres ejes son innecesarios), ya que podemos asignar niveles de excitación y de positividad o negatividad gracias a los datos de los sensores, así como asignar todas las emociones que queramos dentro del círculo.

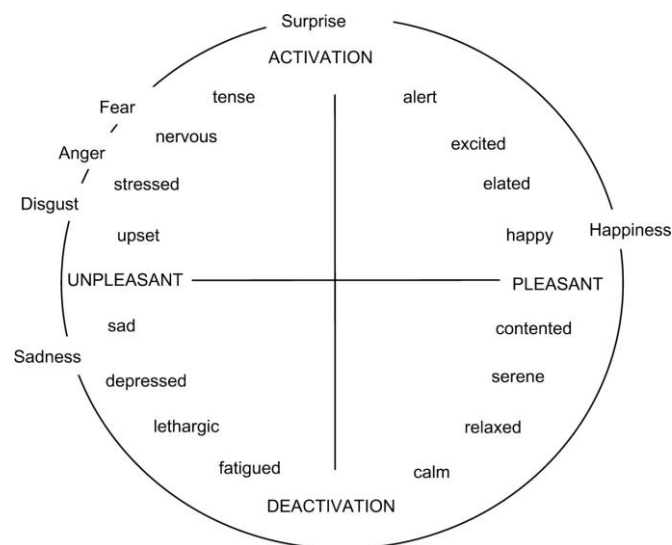


Figura 3.1: modelo circumplejo de Russell basado en dos ejes. Fuente: [6]

Para obtener las coordenadas de la emoción inferida para el modelo de Russell, debemos obtener los valores de cada eje mediante los datos obtenidos de los sensores. Con la lista de sensores más comunes obtenida en la tabla 3.3 y con las conclusiones mostradas después de ella, se ha confeccionado una lista de los posibles datos que se pueden extraer de los sensores y que pueden ser asignados a uno de los dos ejes (actividad o excitación y valencia):

- El acelerómetro y el sensor de proximidad pueden aportar datos sobre la excitación del usuario, como el vestible de *Empatica* [13]. Además el acelerómetro puede proporcionar datos sobre si un movimiento es positivo o negativo, por lo que también aportará información al eje de la valencia.
- El sensor de luz puede aportar información sobre la actividad del usuario si se registran abundantes cambios de luminosidad, lo que a priori indicaría que el usuario está cambiando de lugar continuamente. Además se puede aportar valencia negativa si se tiene en cuenta que el usuario está en lugares lúgubres más tiempo del que suele estar.
- La cámara puede usarse para detectar facciones positivas o negativas, lo que aporta datos a la valencia. También puede determinar baja actividad si se detectan caras de cansancio o de aburrimiento [2].
- El micrófono podrá ser utilizado para medir el tono de voz, lo que aportará datos sobre emociones positivas o negativas [5]. También puede aportar datos de actividad, si el usuario está hablando alto o bajo corresponderá con alta o baja actividad. Por otra parte, se puede obtener lo que está diciendo el usuario y determinar si tiene connotaciones positivas o negativas o si está hablando más de lo normal para él.
- Por último, se puede intentar medir el campo magnético alrededor del usuario y comprobar si afecta a la valencia de las emociones, por ejemplo si es irregular u homogéneo [7].

Teniendo en cuenta estas conclusiones sobre los posibles datos que pueden ofrecer los sensores con interés emocional, se ha creado una tabla a modo de resumen con el valor que aportará a cada eje cada uno de estos sensores:

Sensor	Actividad/Excitación	Valencia
Acelerómetro	Alta actividad si se registran muchos movimientos bruscos. Baja en caso contrario.	Negativa si se detectan temblores, por ejemplo pueden significar miedo o nervios.
Proximidad	Alta actividad si hay muchos cambios de proximidad.	
Luz	Alta actividad si se registran muchos cambios de luminosidad.	Positiva si hay mucha luz, negativa en caso contrario.
Cámara	Baja actividad si se detectan ojos cerrados continuamente (aburrimiento, cansancio)	Positiva si se detectan sonrisas.
Micrófono (qué se dice)	Alta actividad si se registran muchas palabras, significaría que está hablando mucho	Negativa si se registran palabras malsonantes.
Micrófono (cómo se dice)	Alta actividad si se registran muchos gritos (positivos o negativos)	Negativa si se registra un tono irregular de voz
Campos magnéticos		Negativa si el campo magnético es irregular

Tabla 3.4: datos que pueden ofrecer los sensores a los ejes del modelo de Russell. Fuente: propia.

3.1.4 Análisis de requisitos

En este análisis se detallarán todos los requisitos que presenta el desarrollo de esta aplicación. Se dividirá en dos secciones: requisitos funcionales, donde se describirán las funcionalidades que ofrecerá la aplicación y requisitos no funcionales, donde se describirán los criterios que debe cumplir la aplicación.

3.1.4.1 Requisitos funcionales

En este apartado se describirán los requisitos funcionales tanto del apartado interno y lógico de la aplicación, como del apartado externo y disponible para el usuario.

Requisitos lógicos de la aplicación:

RF1: El sistema debe implementar funciones de recogida de datos de todos los sensores disponibles.

- En función de los sensores provistos por el dispositivo en el que se está ejecutando la aplicación, el sistema recogerá los datos interesantes de carácter emocional.
- Para cada sensor se deben recoger datos de la forma adecuada.
- La lista de sensores debe ser extensible para poder añadir nuevos sensores en el futuro.

RF2: El sistema debe implementar un motor de inferencia emocional.

- La aplicación debe implementar un clasificador de emociones que determine la emoción respecto a los valores ofrecidos por los sensores.
- Este motor podrá ser intercambiado por nuevos motores que determinen emociones de distinta forma.

RF3: El sistema debe proporcionar un sistema de persistencia de los datos recogidos por los sensores.

- Debe poderse almacenar el nombre del sensor y sus datos de interés emocional.

RF4: El sistema permitirá recoger datos de los sensores sin tener que tener el dispositivo desbloqueado y la aplicación abierta.

- Deben poder recogerse datos también mediante un servicio en segundo plano transparente al usuario.

RF5: El sistema debe responder de alguna manera adecuada a la emoción del usuario.

- El sistema deberá lanzar al menos notificaciones relacionadas con el estado del usuario cada cierto tiempo.

Requisitos de interacción con el usuario:

RF6: El usuario podrá activar y parar cualquier sensor para la recogida de datos y comprobar su funcionamiento.

- El usuario observará los valores tomados por el sensor que tengan valor emocional.

RF7: El usuario podrá lanzar y parar en cualquier momento el sistema de ejecución en segundo plano (RF4).

RF8: El usuario podrá configurar los valores con los que se lanzará el servicio de recogida de datos en segundo plano.

- Debe poder elegir los sensores que recogerán datos de entre los disponibles.
- Debe poder elegir el tiempo en el que la aplicación interactuará con él (RF5).

RF9: El usuario podrá ver los datos registrados y almacenados por la aplicación.

- Podrá ver una lista con el histórico de los datos registrados: nombre del sensor y señales recogidas (RF3)

RF10: El usuario podrá ver la emoción inferida actualmente y su representación en el modelo clasificador de emociones (RF2).

- Podrá ver los datos tomados para la inferencia de la emoción.
- Podrá ver una representación gráfica de la emoción

RF11: El usuario podrá salir de la aplicación en cualquier momento.

- Si el servicio de recogida de datos está activado, seguirá funcionando en segundo plano.

3.1.4.2 Requisitos no funcionales

La aplicación desarrollada también deberá cumplir los siguientes requisitos no funcionales para que los funcionales sean mejor aprovechados.

RNF1: La aplicación debe poder ser utilizada por cualquier persona capaz de utilizar un Smartphone.

RNF2: La aplicación funcionará en cualquier dispositivo Android provisto de sensores.

- Si el dispositivo no tiene todos los sensores implementados, simplemente no recogerá datos de dicho dispositivo.

RNF3: Todas las opciones que pueda tomar el usuario deben estar distribuidas en un sencillo menú gráfico, con el que se puede interactuar tocando la pantalla.

- Ejecución de recogida de datos de cada sensor y lanzamiento/parada del servicio deben ser accesibles mediante botones pulsables.
- Elecciones de los sensores para el servicio deben implementarse mediante *checkbox*.
- Especificación del tiempo de interacción de la aplicación con el usuario deberá implementarse como caja de texto a la que poder introducir números.

RNF4: Las notificaciones lanzadas por la aplicación deben ser claramente visibles.

- Además de poder verse, el dispositivo también debería lanzar una pequeña vibración.

RNF5: Los datos que pueda almacenar la aplicación serán ilimitados.

- Solo serán limitados por la propia capacidad de almacenamiento del dispositivo.
- Adicionalmente, estos datos no se podrán perder, excepto si el usuario los borra mediante el ajuste de aplicaciones de Android.

RNF6: Todas las operaciones realizadas por el usuario deberán responder de forma inmediata.

- Cada acción implementada mediante botones responderá en menos de 2 segundos.

RNF7: El sistema informará de todos los fallos que se puedan producir, tanto por las acciones del usuario como por la propia lógica del programa.

RNF8: Si el menú desplegado no cabe en su totalidad en la pantalla del dispositivo, se permitirá mover verticalmente para poder ver todas las funcionalidades.

3.1.5 Análisis de tecnologías

3.1.5.1 Tecnologías hardware

Para la implementación de la aplicación definida en los requisitos anteriormente enunciados se requerirá definir el hardware al que irá destinada. Esto es algo que ya se ha explicado en la sección *3.1.1 Análisis de los Smarthpone*, así que en esta sección simplemente se mencionará el dispositivo sobre el que se harán las pruebas y el hardware que posee, esto último se obtendrá por medio de la aplicación gratuita Sensor Box.

El dispositivo para el que se desarrollará la aplicación es el NOAIN A909, incluido en la tabla 3.3, dado que es uno de los que más sensores posee, y que tiene memoria y procesador suficientes para ejecutar la aplicación. Sus características hardware interesantes para esta aplicación son las siguientes:

- Pantalla de 5". Resolución 1280x720
- Acelerómetro BMMA050 de 3 ejes de Bosch. Rango máximo: 32.

- Sensor de luz cm36283 de Capella. Rango máximo: 10240.
- Sensor de orientación BMM050 de Bosch. Rango máximo: 360.
- Sensor de proximidad cm36283 de Capella. Rango máximo: 1.
- Sensor de campos magnéticos BM050 de 3 ejes de Bosch. Rango máximo: 600.
- Cámara frontal (2MP) y trasera (12MP).
- Micrófono.
- GPS.
- Bluetooth.
- WIFI
- Micro USB

Este hardware es más que suficiente para poder ejecutar la aplicación definida en los requisitos. Adicionalmente, también dispone de conexiones por Bluetooth, WIFI y micro USB para poder conectar más sensores.

3.1.5.2 Tecnologías software

Los recursos software que utilizará esta aplicación están determinados por la propia plataforma para la que será desarrollada: Android. Esto sitúa a Java como el lenguaje de programación que se debe usar para la programación del sistema y XML como lenguaje para la creación de las vistas gráficas de la aplicación. También se podría utilizar el potente motor Unity como entorno en el que crear la aplicación, pero dado que la apariencia gráfica de la aplicación no es demasiado importante y por tener más experiencia programando para Android en entornos que utilizan Java y XML, se ha optado por esta opción más simple.

Para cumplir con los objetivos definidos en la sección 3.1.3 para el aprendizaje de la aplicación sobre el usuario, se necesitará un sistema para guardar los datos de los sensores. Para realizar esto, Android nos ofrece tres formas principales, con las que ya se tiene experiencia previa de desarrollo: bases de datos, ficheros y preferencias.

Las preferencias quedan descartadas, porque aunque puedan guardar datos de forma permanente, están pensadas para guardar pequeñas informaciones sobre las preferencias del usuario en una aplicación y sería costoso realizar búsquedas sobre ellas.

Si comparamos los ficheros con las bases de datos observamos que estas últimas ofrecen una mejor solución a este problema, por lo que serán finalmente las elegidas: almacenar datos en un fichero quizá sea más fácil de implementar, pero las bases de datos son mucho más eficientes. Además, su estructura es más lógica y pueden ser modificadas para almacenar los datos de distintas formas y relacionarlos entre sí. Por último, las bases de datos se pueden utilizar para realizar consultas específicas que en una implementación mediante ficheros sería imposible.

Android incorpora la librería SQLite que nos permitirá utilizar bases de datos mediante el lenguaje SQL, de una forma sencilla y utilizando muy pocos recursos del sistema. A primera vista, la base de datos que se creará ni siquiera deberá seguir el modelo relacional, puesto que no habrá relaciones y nada más que una tabla, la que se usará para guardar el nombre del sensor y los datos que recoja. Esta tabla nos permitirá recuperar todas las señales recogidas por todos los sensores y mediante consultas SQL, los datos de un sensor en concreto a través de su nombre.

3.2. Diseño

Un software completo diseñado para la computación afectiva debería estar compuesto al menos por un motor de inferencia y otro de expresión:

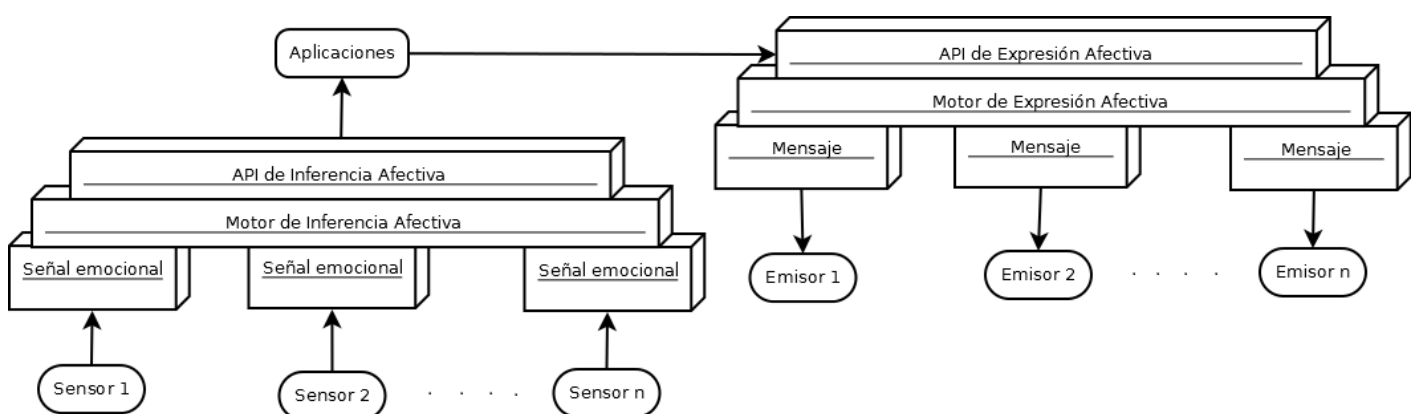


Figura 3.2: Modelo de Software afectivo. Fuente: propia

El motor de inferencia de un software afectivo debe proporcionar un sistema para determinar el estado emocional de los usuarios. Para ello, debe estar compuesto de al menos los siguientes elementos:

- Sensores: debe disponer de varios para poder acceder a las señales que se puedan capturar de ellos y que hará llegar al motor de inferencia afectiva. Pueden ser las cámaras o micrófonos de los dispositivos, por ejemplo.
- Motor de inferencia afectiva: gracias a las señales provistas por los sensores, este elemento deberá implementar mecanismos para determinar y clasificar el estado emocional del usuario.

Esto se podrá utilizar para desarrollar aplicaciones que conozcan la emoción por la que está pasando el usuario y actuar de una mejor forma con él, momento en el que entrará en acción el segundo motor, el motor de expresión afectiva. Este motor es el encargado de hacer que la aplicación/dispositivo exprese emociones, para ello deberá disponer de los siguientes elementos:

- Motor de expresión afectiva: mediante la emoción del usuario obtenida por el motor de inferencia afectiva o por otras entradas definidas por el desarrollador, deberá ofrecer un mecanismo para expresar una emoción adecuada en respuesta a ella. Para ello hará uso de los emisores disponibles, ejecutando un mensaje en cada uno de ellos.
- Emisores: son utilizados en conjunto para expresar la emoción que se desea interpretar gracias a las órdenes enviadas por el motor de expresión afectiva. Estos elementos son las pantallas, altavoces y sistemas de vibración, entre otros.

El diseño de la aplicación de este trabajo se centra en el primero de estos motores, el descrito en la parte izquierda de la figura 3.2. Este diseño se debe dividir en dos partes: la biblioteca que permitirá utilizar los sensores e inferir la emoción a partir de ellos y la aplicación que hará uso de las funciones de esta biblioteca. Todo ello se diseñará para ser implementado en un lenguaje orientado a objetos, concretamente en Java.

3.2.1 Diseño de la biblioteca

La biblioteca necesita una entidad principal: el sensor. Deberá manejar varias de estas entidades y ser extensible para poder añadir nuevos sensores sin problema. Las funcionalidades principales de la entidad sensor serán recoger datos de los sensores de los dispositivos y almacenarlos en otro lugar, que será la base de datos. Los

sensores que formarán parte de la primera versión de la aplicación serán los siguientes:

Acelerómetro, cámara, campo magnético, luminosidad, micrófono, proximidad y voz

Por otra parte, la biblioteca también estará formada por un motor de inferencia, el cual será capaz de calcular la emoción actual gracias a los datos almacenados y que podrá ser intercambiado por nuevos motores. Como se comentó en la sección de análisis, el clasificador de emociones que se implementará será el propuesto por Russell [6], basado en la excitación y la valencia de las señales. El valor de las señales irá desde un valor de -100 hasta 100 para los dos ejes, por lo que no será un círculo como en algunas representaciones de Russell, si no un cuadrado. Una señal positiva estará en la parte derecha del eje, mientras que una señal negativa estará en la parte izquierda, por otro lado la detección de una alta actividad dará un alto valor al eje y, poca actividad dará un valor negativo. La recogida de los datos, el cálculo de la emoción a partir de ellos y su representación gráfica se explicarán con detalle en la sección Implementación.

Para hacer más específico el reconocimiento, se situarán 27 emociones distintas definidas con una zona recuadrada cada una definiendo dónde empieza y termina su x y su y. Las zonas cercanas al origen proporcionarán una emoción indeterminada. Estas emociones principales se distribuirán en los cuatro cuadrantes y serán las de la siguiente figura:

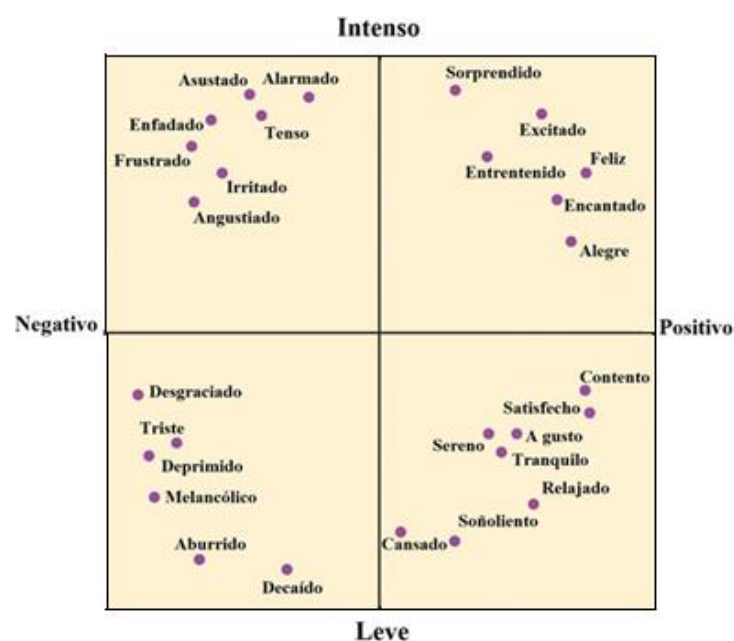


Figura 3.3. Emociones distribuidas en el modelo de Russell. Fuente: [6] y propia.

El prototipo/maqueta que muestra la navegación y funcionamiento de la primera versión de la aplicación será la siguiente:

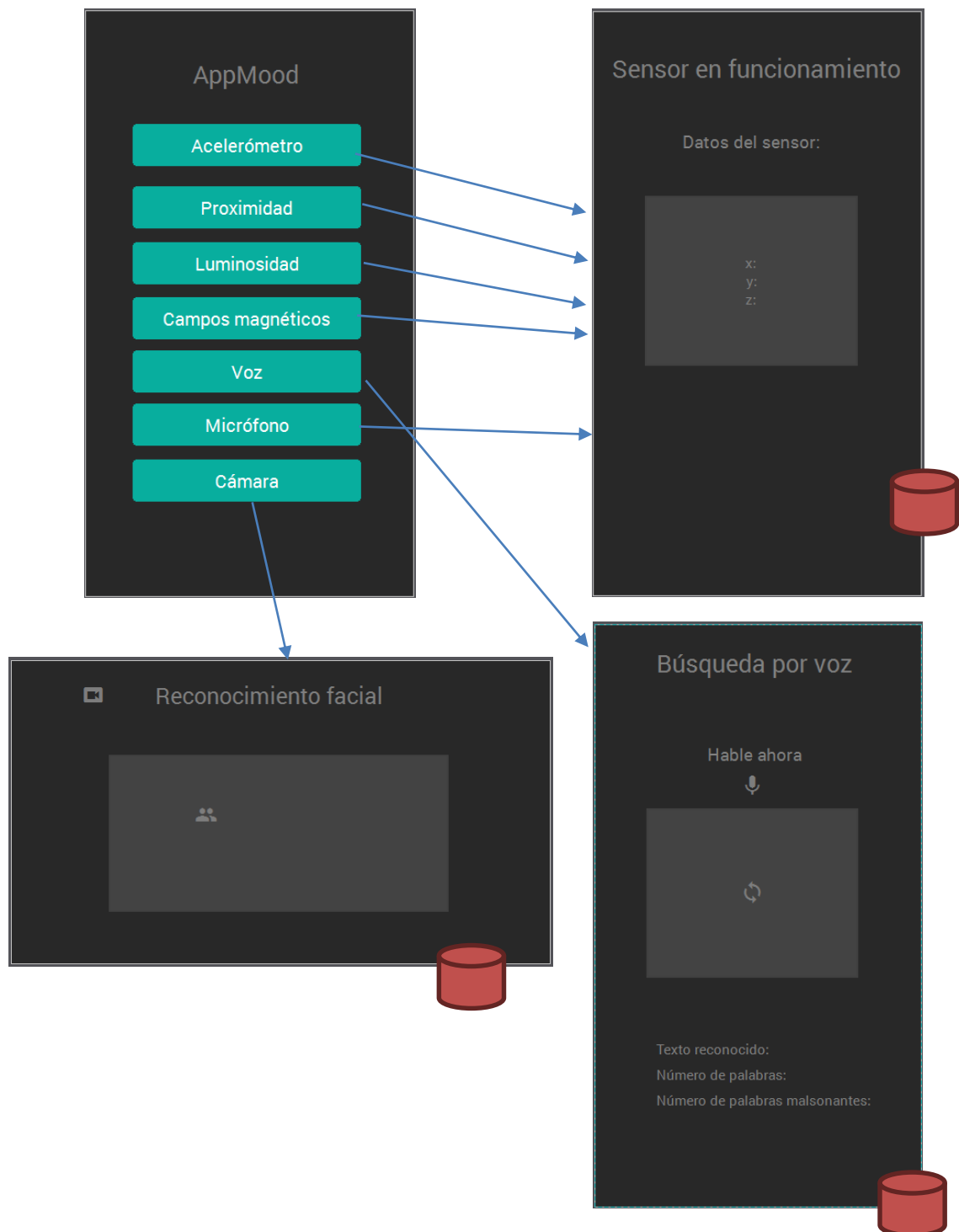


Figura 3.4. Maqueta de la primera versión de la aplicación.

La primera versión de la aplicación se utilizará para comprobar gráficamente el funcionamiento de los sensores y poder determinar los que serán adecuados para su

uso en la aplicación de prueba de la biblioteca y que serán utilizados por el motor de inferencia para determinar la emoción del usuario. Todos estos sensores mostrarán una pequeña ventana donde aparecerán los valores que están capturando en cada momento y que sean de interés emocional. El caso de la cámara y de la voz serán distintos porque usarán bibliotecas externas que recogerán y mostrarán los datos de distinta forma. Por ejemplo, la cámara mostrará lo que se está viendo en ese momento y los gestos emocionales de las caras que encuentre.

3.2.2 Diseño de la aplicación AppMood

La aplicación final hará uso de las entidades y funciones definidas en la biblioteca, pero además añadirá un sistema de recogida de datos en segundo plano, el cual hará uso de los sensores que puedan funcionar de esta forma, y los añadirá a la base de datos.

También añadirá un sistema de interacción con el usuario: cada cierto tiempo, elegible por el usuario, el sistema mandará una notificación sobre el estado en el que ha considerado que se siente el usuario y reproducirá automáticamente una canción introducida previamente acorde a ese estado emocional.

Para poder implementar las funcionalidades de la aplicación final estudiaremos todas las acciones que el usuario debería poder realizar, definidas en la sección de requisitos y realizaremos un diagrama simple de casos de uso, que se puede observar en la siguiente figura, la 3.5.

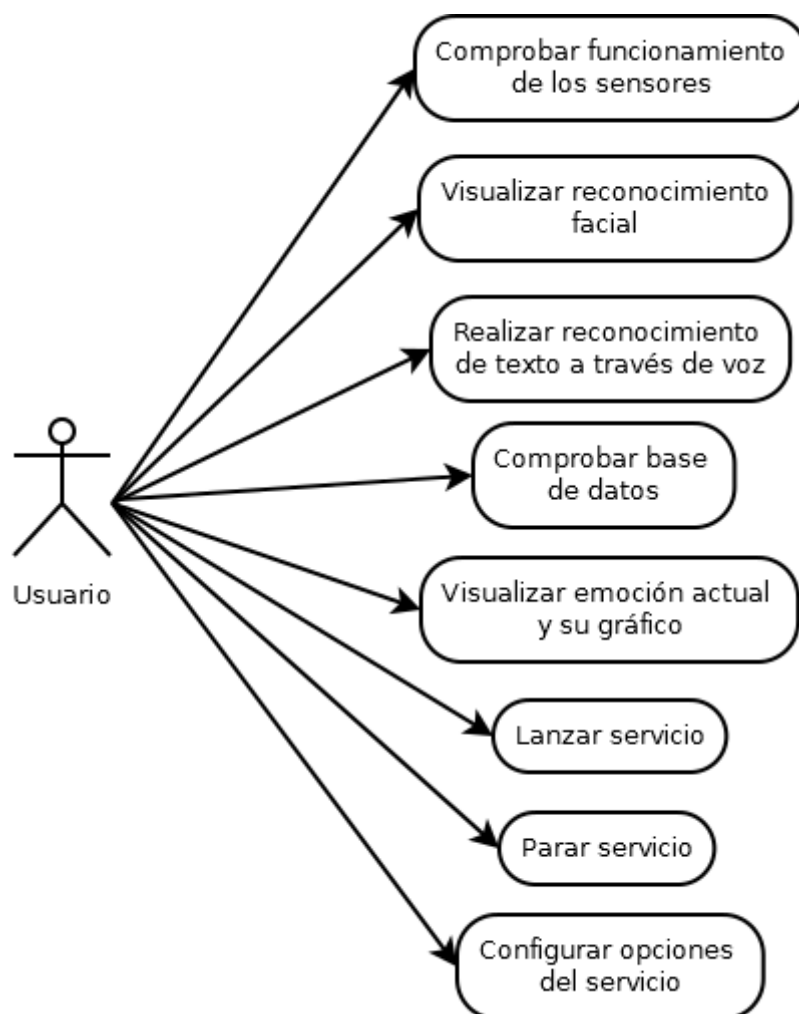


Figura 3.5: Casos de usos de la aplicación AppMood

El único actor de la aplicación será el usuario y podrá realizar las siguientes acciones:

- Comprobar funcionamiento de los sensores: el usuario podrá acceder a cada sensor y ponerlo a funcionar, observando los cambios continuos que el sensor muestra. Podrá pararlo cuando quiera, momento en el cual, la aplicación guardará en la base de datos los datos interesantes que haya recogido.
- Visualizar reconocimiento facial: el usuario podrá acceder al sistema de reconocimiento facial y observar las imágenes recogidas a través de la cámara junto a los recuadros marcados por la aplicación para señalar movimientos interesantes emocionalmente.
- Realizar reconocimiento de texto a través de voz: el usuario podrá acceder al sistema de reconocimiento de su propia voz. La aplicación utilizará el sistema de “búsqueda por voz” proporcionado por Google y mostrará el texto

reconocido de la voz del usuario, así como el número de palabras y palabras malsonantes.

- Comprobar base de datos: el usuario podrá visualizar todos los datos registrados en la base de datos, que consistirán en una recopilación de todas las señales interesantes recogidas por el sensor.
- Visualizar emoción actual y gráfico: el usuario podrá obtener la emoción calculada actualmente gracias a los últimos datos registrados, además de su posición en el gráfico valencia-excitación definido por Russell y explicado en el análisis.
- Lanzar servicio: el usuario podrá lanzar el servicio de recogida de datos en cualquier momento.
- Parar servicio: el usuario podrá parar el servicio lanzado previamente.
- Configurar opciones del servicio: el usuario podrá configurar los sensores funcionales en el servicio, así como el tiempo de interacción con el usuario.

La aplicación final estará diseñada según este diagrama de clases, en el que se incluyen las funciones de la biblioteca y de la aplicación:

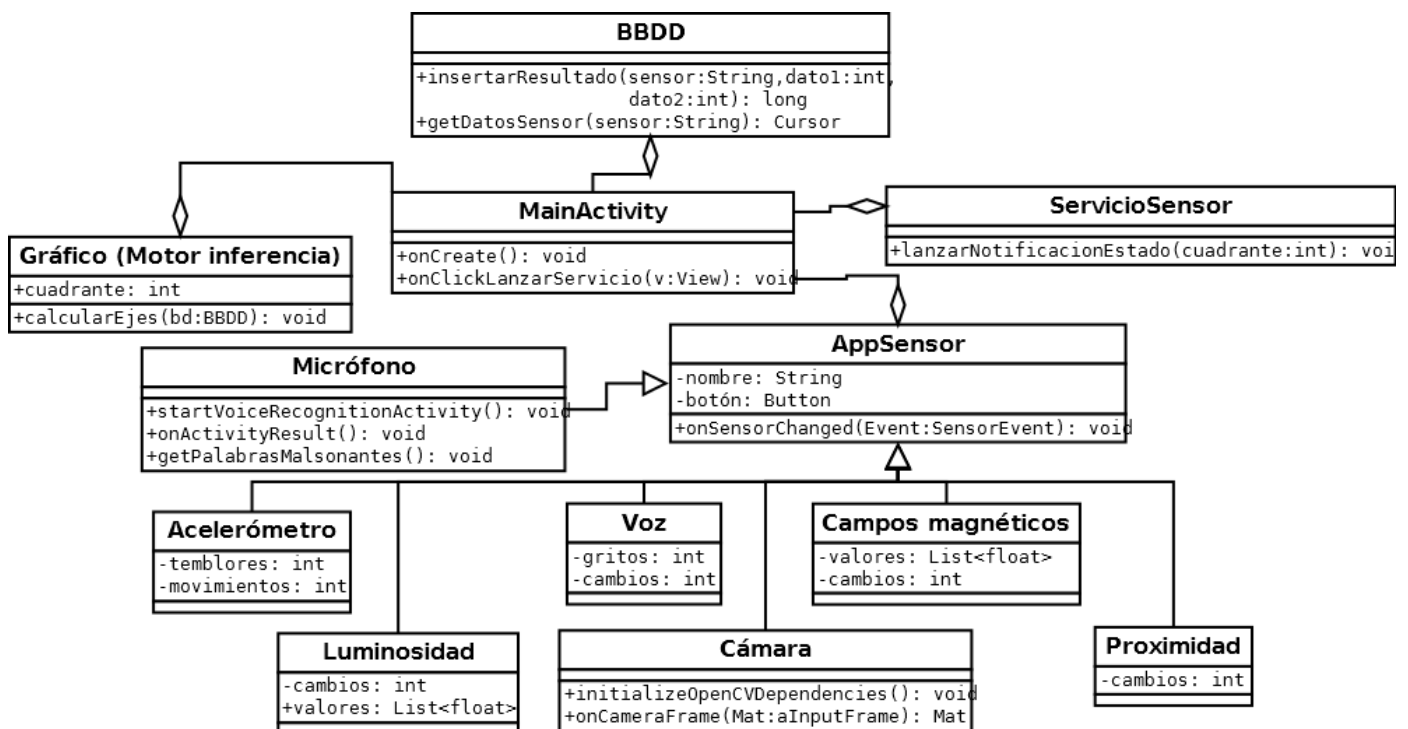


Figura 3.6: diagrama de clases simplificado de la aplicación AppMood.

Todas estas clases y sus relaciones serán ampliamente detalladas en la siguiente sección, Implementación.

El menú de la aplicación AppMood añadirá las siguientes funcionalidades y pantallas a la maqueta de la primera versión, descrita en la figura 3.3:

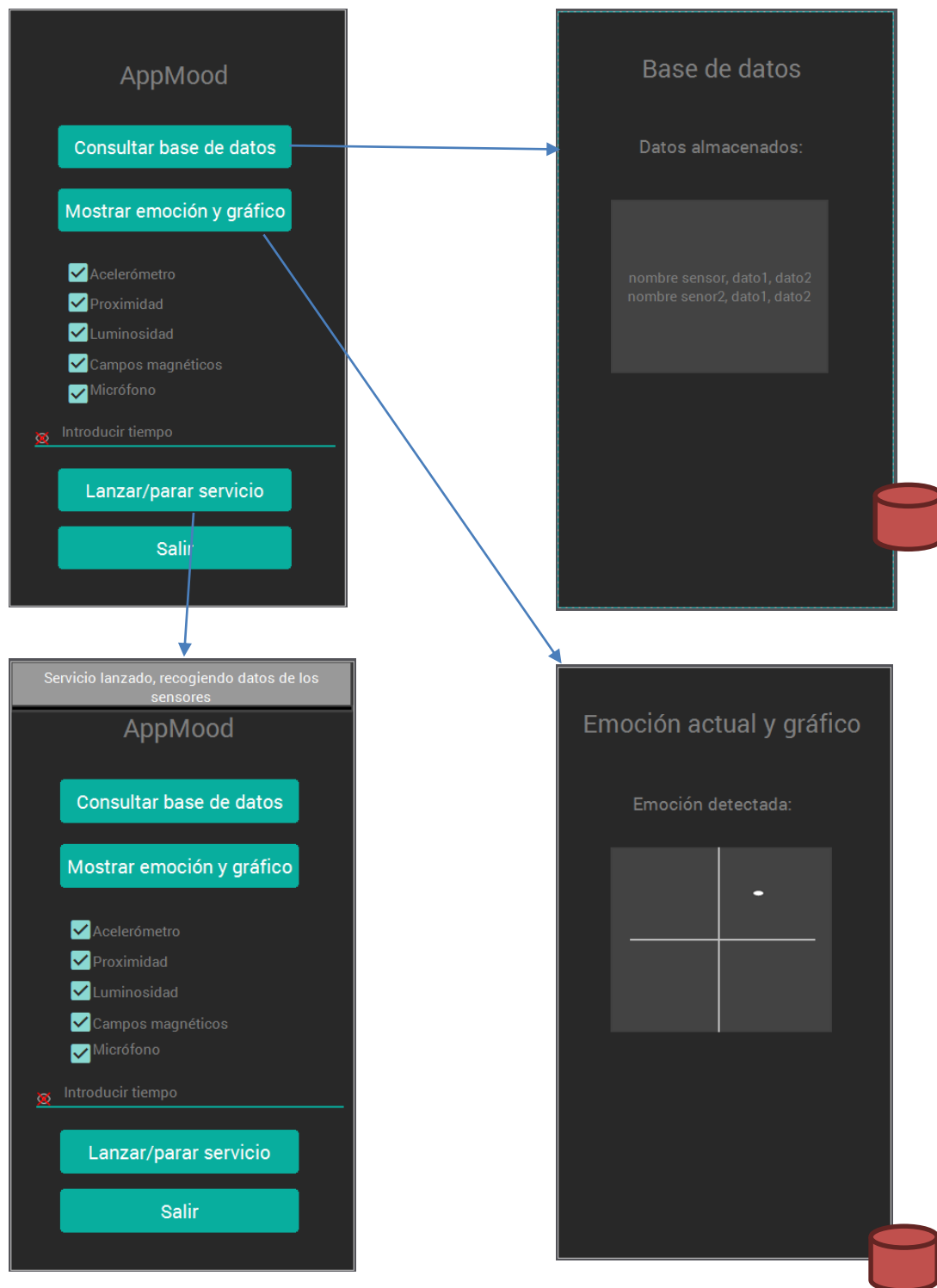


Figura 3.7: maqueta de las funcionalidades añadidas por la aplicación AppMood.

3.3. Implementación

En esta sección se explicará detalladamente la implementación del diseño de la aplicación descrito en la sección anterior: las clases, sus métodos y atributos y la relación entre ellas.

El lenguaje en el que se implementará la lógica de la aplicación será Java, mientras que el lenguaje para diseñar gráficamente las pantallas será XML, como ya se comentó en la sección 3.1.5.2 Tecnologías Software. El entorno de desarrollo para realizar la implementación será Eclipse mediante el uso del SDK de Android. Las ventajas de la utilización de este entorno radican en la experiencia previa, la popularidad y la facilidad para poder incluir librerías externas.

3.3.1 Implementación de la biblioteca

Para implementar las funcionalidades principales de la biblioteca, la clase principal que se ha de crear es una que sea capaz de representar a un sensor. Esta clase deberá al menos contener el nombre del sensor y un método para la recogida de señales interesantes. Como se puede observar en la figura 3.6, esta clase será `AppSensor`, y de ella heredarán otras siete clases, que serán las que representen a cada sensor y recogerán los datos de la forma adecuada cada una, los atributos de estas clases contendrán las señales interpretadas con interés emocional, definidas en la figura 3.4.

`AppSensor.java`, extiende de la clase `Activity` para poder usarse como una actividad Android y poder añadirle una vista (view) XML para visualizar los datos recogidos.

- **Atributos**

- Nombre, que identificará unívocamente al sensor.
- Botón, button que se utilizará para poder activar o parar la recogida de datos.

- **Métodos**

- `CalcularMedia(list<float>)`, se utilizará para realizar la media de una lista de datos.
- `CalcularModulo(float x, float y, float z)`, se utilizará para calcular el módulo de un vector.

Las siete clases que heredarán de AppSensor se muestran a continuación, la mayoría implementarán la interfaz SensorEventListener proporcionada por Android y que se utiliza para detectar cambios captados por los sensores mediante el método onSensorChanged(SensorEvent Event).

Un método común a todas estas clases será onCreate(), que será el método llamado al crear el sensor. Activará y desactivará la recogida de datos mediante el toque del botón por parte del usuario. Si el usuario desactiva la recogida de datos, en ese momento se almacenarán en la base de datos las señales recogidas.

Acclerometro.java, extiende de la clase AppSensor e implementa la interfaz SensorEventListener. Se utiliza para recoger datos del acelerómetro e interpretarlos como señales emocionales.

- **Atributos**

- Movimientos_bruscos, número entero que contendrá las veces que se ha agitado bruscamente el dispositivo. Aportará alta actividad al modelo de Russell.
- Temblores, número entero que indicará los temblores en la mano del usuario. Aportará negatividad al modelo de Russell.
- Tiempo, número real utilizado para la contabilización de temblores. Establecido en 200 milisegundos.
- LimiteTemblor, número real utilizado para la contabilización de temblores. Establecido como 0.8.

- **Métodos**

- onSensorChanged(SensorEvent Event), sobrescrito para recoger datos del acelerómetro de la forma más interesante desde el punto de vista emocional y representarlos al usuario. Este sensor nos ofrece la aceleración en x, y y z. En la sección Tecnologías Hardware observamos que el valor máximo del acelerómetro es 32, por lo que deberemos considerar que si uno de los ejes sobrepasa un valor de 20, se ha realizado un movimiento brusco, lo que incrementará el contador de movimientos_bruscos. Por otra parte, si en un momento dado se registra un valor, y rápidamente (en menos tiempo que el proporcionado por el atributo tiempo) se registra el mismo valor pero con una cifra constante (el limiteTemblor) más o menos, se aumentará el contador de temblores.



Figura 3.8. Acelerómetro en funcionamiento

Luminosidad.java, extiende de la clase `AppSensor` e implementa la interfaz `SensorEventListener`. Se utiliza para recoger datos de la luz ambiental e interpretarlos como señales emocionales.

- **Atributos**

- `Numero_cambios`, número entero que contendrá las veces que se ha registrado un cambio de luz. Aportará alta actividad al modelo de Russell.
- `Valores`, lista de número reales que indicará todos los valores recogidos. La media de ellos aportará positividad o negatividad al modelo de Russell.

- **Métodos**

- `onSensorChanged(SensorEvent Event)`, sobrescrito para recoger datos del sensor de luminosidad de la forma más interesante desde el punto de vista emocional y representarlos al usuario. Este sensor nos ofrece el valor de la luz actual en lúmenes. En cada cambio de luz añadirá este valor a la lista de valores y calculará su media, para obtener la luz media de la sesión. Cada cambio también aumentará el atributo `numero_cambios`.



Figura 3.9. Sensor de luminosidad en funcionamiento

Proximidad.java, extiende de la clase `AppSensor` e implementa la interfaz `SensorEventListener`. Se utiliza para recoger datos del sensor de proximidad e interpretarlos como señales emocionales.

- **Atributos**

- `Numero_cambios`, número entero que contendrá las veces que se ha registrado un cambio de proximidad, este valor será 1 o 0 si hay algo próximo al dispositivo o no. Aportará alta actividad al modelo de Russell.

- **Métodos**

- `onSensorChanged(SensorEvent Event)`, sobrescrito para recoger datos del sensor de proximidad de la forma más interesante desde el punto de vista emocional y representarlos al usuario. Este sensor nos ofrece el valor de si hay algo próximo o no al dispositivo, como se vio en la sección Tecnologías Hardware, este límite será de 1cm. En cada cambio de proximidad se aumentará el atributo `numero_cambios`.



Figura 3.10. Sensor de proximidad en funcionamiento

CampoMagnetico.java, extiende de la clase AppSensor e implementa la interfaz `SensorEventListener`. Se utiliza para recoger datos del sensor de campos magnéticos e interpretarlos como señales emocionales.

- **Atributos**

- `Numero_cambios`, número entero que contendrá las veces que se ha registrado un cambio brusco en el campo magnético, lo que indicará que no es constante. Aportará negatividad al modelo de Russell.
- `Valores`, lista de número reales que indicará todos los valores recogidos. La media de ellos aportará actividad al modelo de Russell.
- `Tiempo`, número real utilizado para la contabilización de cambios. Establecido en 100 milisegundos.
- `LimiteCambio`, número real utilizado para la contabilización de cambios. Establecido como 1.1.

- **Métodos**

- `onSensorChanged(SensorEvent Event)`, sobrescrito para recoger datos del sensor de campos magnéticos de la forma más interesante desde el punto de vista emocional y representarlos al usuario. Este sensor nos ofrece el campo magnético en x, y y z. En la sección Tecnologías Hardware observamos que el valor máximo del campo magnético es 600. Si en un

momento dado se registra un valor, que se calculará mediante el módulo de los tres valores proporcionados, y rápidamente (en menos tiempo que el proporcionado por el atributo tiempo) se registra el mismo valor pero con multiplicado por el limiteCambio se considerará un campo magnético irregular y no homogéneo, lo que aumentará el atributo numero_cambios.



Figura 3.11. Sensor de campos magnéticos en funcionamiento

Camara.java, extiende de la clase `AppSensor` e implementa la interfaz `CvCameraViewListener`. Se utiliza para recoger datos de la cámara e interpretarlos como señales emocionales. Esta clase en lugar de actuar como los sensores anteriores, necesitará una librería externa, comentada en el Estado del Arte sobre reconocimiento facial: `openCV`. Esta librería nos permitirá acceder a la cámara y reconocer formas y objetos, en nuestro caso se utilizará para reconocer caras, sonrisas y ojos. Este sensor no será utilizado por la aplicación `AppMood` puesto que no puede funcionar en servicio y su uso se reduce a las ocasiones en las que tenga la cara del usuario delante de la cámara. De todos modos se ha incluido para demostrar que la cámara de los Smartphone es útil a la hora de detectar emociones, en casos como videoconferencias, por ejemplo. Es importante recalcar que necesita la librería `openCV` instalada en el dispositivo para poder funcionar.

- **Atributos**

- Numero_sonrisas, número entero que contendrá las veces que se ha registrado una sonrisa en una cara. Aportará positividad al modelo de Russell.
- Número_ojos_cerrados, número entero que contendrá las veces que se no se reconocen ojos abiertos en una cara reconocida. Aportará actividad al modelo de Russell.

- **Métodos**

- initializeOpenCVDependencies(), proporcionado por la interfaz CvCameraViewListener, se usa para añadir los clasificadores de caras, sonrisas y ojos mediante ficheros xml entrenados proporcionados por la propia biblioteca.
- onCameraFrame(Mat aInputFrame) sobrescrito para recoger la imagen captada por la cámara del dispositivo y dibujar rectángulos alrededor de las caras que encuentre (limitado a una cara) y sobre los ojos y sonrisas dentro de ellas. Para ello comparará si los datos incluidos en los ficheros entrenados se parecen a alguna parte de la imagen capturada por la cámara en alguna escala. Aumentará el numero_sonrisas al detectar sonrisas y el numero_ojos_cerrados al perder ojos encontrados, lo que indicaría demasiado aburrimiento, sueño o desgana.

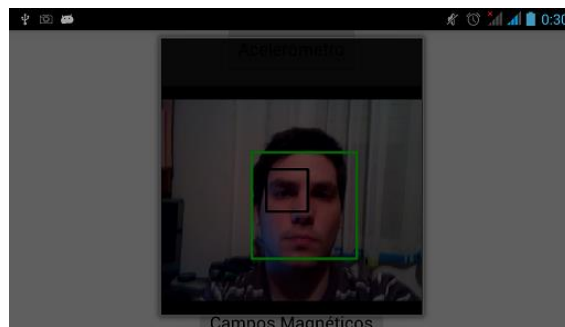


Figura 3.12. Reconocedor facial en funcionamiento

Microfono.java, extiende de la clase AppSensor. Mediante el sistema de búsqueda por voz de Google, se utiliza para recoger las palabras pronunciadas por el usuario e interpretarlas como señales emocionales. Este “sensor” tampoco se ha incluido en la aplicación AppMood por no poder funcionar en modo servicio. De todas formas se incluye aquí para demostrar que se pueden obtener datos emocionales por lo que dice

el usuario y que serían interesantes cuando se puedan recoger estos datos de otro modo en el futuro.

- **Atributos**

- Numero_palabras, número entero que contendrá el número de palabras pronunciadas por el usuario. Aportará actividad al modelo de Russell.
- Malsonantes, número entero que contendrá el número de palabras malsonantes pronunciadas por el usuario. Aportará negatividad al modelo de Russell.

- **Métodos**

- getPalabrasMalsonantes(), rellenará una lista de strings con las palabras malsonantes que se van a reconocer, estas palabras estarán incluidas en un fichero de texto [21].
- startVoiceRecognitionActivity(), comenzará el sistema de búsqueda por voz de Google para recoger la voz del usuario y traducirla a texto. Se seleccionará español como lenguaje a reconocer.
- onActivityResult(int requestCode, int resultCode, Intent data), recogerá las palabras reconocidos y las contabilizará, añadiendo ese número a numero_palabras. Comprobará una a una si se encuentra en la lista de palabras malsonantes, si están incrementa el atributo malsonantes. Cabe destacar que algunas palabras malsonantes, el reconocedor de Google las devuelve como una cadena de asteriscos, o censurando parte de la palabra, así que si la palabra contiene un asterisco se incrementará el atributo malsonantes sin necesidad de buscarla en la lista (puesto que no va a encontrar una cadena de asteriscos). Finalmente añadirá los dos atributos a la base de datos.



Figura 3.13. Búsqueda por voz en funcionamiento

Voz.java, extiende de la clase AppSensor. Se utiliza para recoger datos del medidor de decibelios e interpretarlos como señales emocionales.

- **Atributos**

- Numero_cambios, número entero que contendrá las veces que se ha registrado un cambio brusco en la voz del usuario, indicando un tono de voz irregular. Aportará negatividad al modelo de Russell.
- Gritos, número entero que indicará los gritos del usuario. Aportará actividad al modelo de Russell.

- **Métodos**, los métodos importantes de esta clase serán los que proporciona una clase interna denominada Ear y que actuará de forma asíncrona.

- doInBackground(), dormirá el sistema durante 300 milisegundos, momento en el cual publicará los datos de la amplitud recogida en decibelios.
- onProgressUpdate(Double value), recogerá el dato publicado por el método anterior, que será el volumen en decibelios, desde -80 a 0. Si se recoge un dato mayor a -5 se contabilizará un grito. Si se registra un dato y el siguiente es mayor o menor en 25 decibelios aumentará el contador de cambios irregulares en la voz.

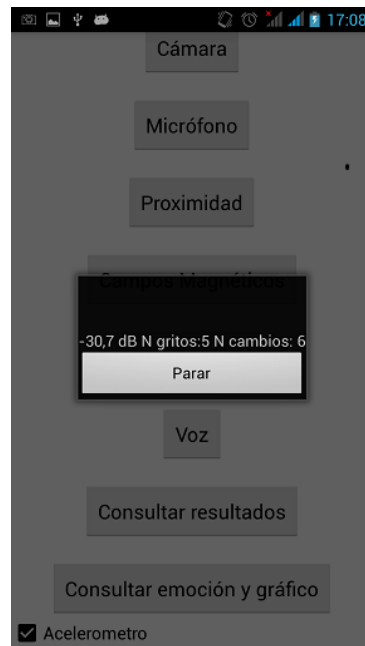


Figura 3.14. Medidor de decibelios en funcionamiento

La otra clase esencial de la biblioteca será la encargada de la persistencia de los datos recogidos por las clases heredadas de AppSensor.

BBDD.java, creará una base de datos mediante una clase interna llamada DatabaseHelper que extenderá de la clase proporcionada por Android para la creación de bases de datos SQLiteOpenHelper. Mediante el context de cualquier actividad se podrá acceder a esta base de datos. La clase **datos.java** se utilizará como activity que representará gráficamente los datos de la base de datos.

- **Atributos**, las cadenas de texto para crear los nombres de las tablas.
- **Métodos**
 - createTable(SQLiteDatabase db), mediante una sentencia sql creará la única tabla que utilizará la base de datos, la tabla *resultados*. Estará compuesta por una clave primaria compuesta por un entero que se autoincrementará, el nombre del sensor y dos enteros que serán suficientes para guardar las señales recogidas por los sensores, ya que ninguno recoge más de dos datos.
 - insertarResultado(string cadena, int dato1, int dato2), mediante una inserción a la base de datos, añadirá los resultados de un sensor a la tabla *resultados*.
 - getResultados(), devuelve todos los resultados guardados en la tabla mediante una query.

- `getDatosSensor(string sensor)`, devuelve todos los resultados de un sensor en concreto guardados en la tabla mediante una query.

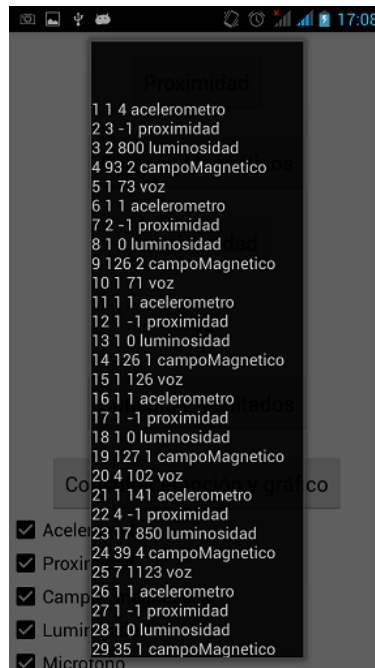


Figura 3.15. Resultados guardados en la base de datos.

La clase que se utilizará para implementar el motor de inferencia será `Gráfico.java` y también se utilizará para mostrar gráficamente la emoción inferida, para ello utilizará la clase que implementa una *View*, **`graph.java`**, que dibujará el eje de coordenadas y situará la emoción mediante un punto.

`Gráfico.java`, extiende de `Activity` y se utiliza para calcular la emoción probable del usuario y representarla en un eje de coordenadas.

- Atributos

- **Excitación**, entero que determina la excitación en el modelo de Russell.
- **Valencia**, entero que determina la valencia en el modelo de Russell.
- **DatosExcitacion**, lista de valores que contiene todos los datos de excitación obtenidos de los sensores.
- **DatosValencia**, lista de valores que contiene todos los datos de valencia obtenidos de los sensores.
- **Cuadrante**, entero que indica el cuadrante en el que se encuentra la emoción inferida.
- **Db**, base de datos de la que sacar los valores.

- **Métodos**

- calcularEjes(BBDD bd), determina los valores de los atributos de excitación, valencia y cuadrante a partir de los datos de una base de datos. Para ello, comparará el último valor recogido por cada sensor (que determine valencia o excitación), con la media de todos los valores recogidos históricamente por ese mismo sensor, y determinar así el porcentaje en el que los últimos datos son mayores o menores a lo normal para el usuario, determinando un valor entre -100 y 100. Después se hará la media de todas estas comparaciones de todos los sensores y se obtendrán los valores de valencia, excitación y el cuadrante en el que se encuentra la emoción. Con estos datos, determinará cuál de las 27 emociones definidas en el diseño será la más adecuada, si ninguna es lo suficientemente cercana será *indeterminada*.



Figura 3.16. Resultado de ejecutar la visualización de resultados del motor de inferencia con datos de ejemplo.

3.3.2. Implementación de AppMood

Además de las clases definidas hasta ahora, la aplicación final añadirá otras nuevas para cumplir con los requisitos apuntados en el análisis. Esta aplicación está pensada para demostrar el uso de un software afectivo de forma práctica, recogerá el estado emocional del usuario en ese momento e interactuará con él de forma adecuada, en este caso el servicio le mandará un mensaje por notificación y reproducirá una canción recomendable para ese estado de ánimo.

MainActivity.java, clase que contiene la actividad principal e inicial de la aplicación, en ella se mostrará todo el menú y permitirá al usuario acceder a las demás actividades mediante botones.

- **Atributos**

- Botones para acceder a las demás actividades de la aplicación.

- **Métodos**

- onCreate(), añade un escuchador a cada uno de los botones para acceder a las demás actividades.
 - onClickLanzarServicio(View v), método que lanza o para el servicio de recogida de datos si el usuario pulsa el botón asignado. Mediante un *intent* lanza el servicio en segundo plano y mediante *extras*, le pasa los sensores que ha elegido lanzar el usuario, además del tiempo de interacción del servicio con el usuario. Mediante checkboxes, el usuario puede elegir cuál de estos cinco sensores van a estar recogiendo datos: acelerómetro, proximidad, luminosidad, campos magnéticos y micrófono, esto es útil porque si el usuario va a llevar el dispositivo en el bolsillo, no tendrá sentido que capture datos de la luz, por ejemplo. Mediante una caja de texto que solo acepta números, el usuario indicará el tiempo de interacción.



Figura 3.17. Apariencia del menú lanzado por MainActivity.java

ServicioSensor.java, clase que extiende de Service para poder implementar un servicio en segundo plano, implementa SensorEventListener para recoger datos de los sensores. Recogerá datos de los sensores y lanzará notificaciones al usuario cada cierto tiempo, además de reproducir una canción adecuada para el estado de ánimo inferido.

- **Atributos**

- Los mismos atributos que usan las clases herederas de AppSensor.
- Canción, MediaPlayer que reproducirá una canción.
- g, instancia de la clase Grafico para la inferencia del estado de ánimo.

- **Métodos**

- onCreate(), lanzará el servicio e inicializará todas las variables. Lanzará un mensaje al usuario informando de que se ha creado el servicio.
- onStart(), lanzará una notificación informando de que el servicio está recogiendo datos. Activará la recogida de datos desde los sensores elegidos por el usuario.
- lanzarNotificacion(), lanza la notificación enviada por onStart().
- lanzarNotificacionEstado(int cuadrante), mediante el valor pasado como parámetro, el servicio lanzará una notificación y reproducirá una canción

adecuada. Para simplificar esto, se han elegido 4 notificaciones y 4 canciones, correspondientes a cada cuadrante del modelo de Russell:

"Estás muy animado, ¿no?"
 "La verdad es que estás bastante relajado"
 "Un día triste, ¿eh?"
 "Tranquilízate, escucha esta canción"

- onSensorChanged(SensorEvent event), recogerá datos de todos los sensores activos de la misma forma que los sensores descritos herederos de AppMood, excepto de la cámara y del reconocimiento de voz, que no pueden funcionar en un servicio. Cuando el tiempo de ejecución pase del tiempo indicado por el usuario, guardará los datos recogidos por los sensores en la base de datos y calculará mediante la clase Gráfico el cuadrante correspondiente a la emoción, para después llamar a lanzarNotificacionEstado con ese valor y seguir recogiendo datos hasta que vuelva a pasar el tiempo correspondiente.

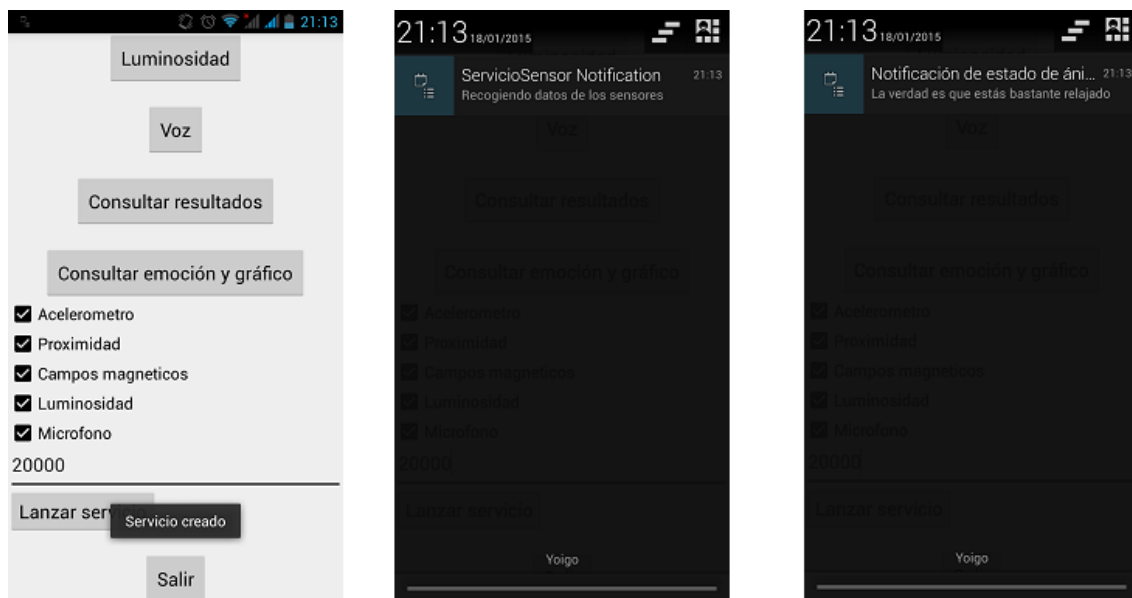


Figura 3.16. Mensajes y notificaciones lanzadas por ServicioSensor.java

3.3.3 Otras consideraciones sobre la implementación (interfaces y recursos)

Además del código java correspondiente a las clases de la biblioteca y de la aplicación, también hay que comentar los recursos añadidos al proyecto. Se han utilizado cuatro ficheros xml para la definición de las interfaces de la aplicación:

- **Activity_main.xml**, define el menú de la aplicación, está formado por el título, los botones para acceder a las demás actividades, los checkbox para elegir los sensores que usará el servicio, una caja de texto para introducir el tiempo y un botón para lanzar/parar el servicio. Usado por la clase MainActivity.
- **Camara.xml**, sirve para mostrar lo que está viendo la cámara y que la clase cámara detecte las señales interesantes emocionalmente.
- **Datos.xml**, sirve para mostrar los datos de la base de datos en forma de lista. Usado por la clase BBDD.
- **Sensor.xml**, sirve para mostrar los datos que está registrando a cada momento el sensor. También está compuesto por un botón para comenzar y parar la recogida de datos. Usado por todos los sensores excepto por la cámara y la voz.

Por otra parte también es necesario comentar los recursos añadidos a la carpeta del proyecto *raw* para el correcto funcionamiento de la aplicación:

- Cuatro canciones en formato mp3, cada una correspondiente a un cuadrante del modelo de Russell.
- Un fichero.txt con una lista de palabras malsonantes, utilizado por la clase micrófono.
- Tres ficheros xml entrenados para detectar caras, ojos y sonrisas, utilizados por la clase cámara.

CAPÍTULO 4

PRUEBAS

En este capítulo se explicarán las pruebas a las que ha sido sometido el software generado, mostrando sus resultados y explicando si son los esperados. Para resumir la batería de pruebas a las que se ha sometido la aplicación, sencillamente se mostrarán los resultados de las pruebas unitarias y de sistema mediante caja negra, esto es, comprobando si los resultados son los esperados mediante las acciones del usuario.

Lamentablemente, el sistema no se ha podido probar con más usuarios reales que el propio desarrollador, debido al tiempo que necesita la aplicación para aprender y recoger datos del usuario.

4.1. Pruebas unitarias

Estas pruebas corresponden a las realizadas a cada sensor implementado.

Prueba nº1: comprobar que el acelerómetro recoge correctamente datos de movimientos bruscos. Para ello se ha puesto en marcha dicho sensor y se ha agitado el dispositivo violentamente durante un tiempo.

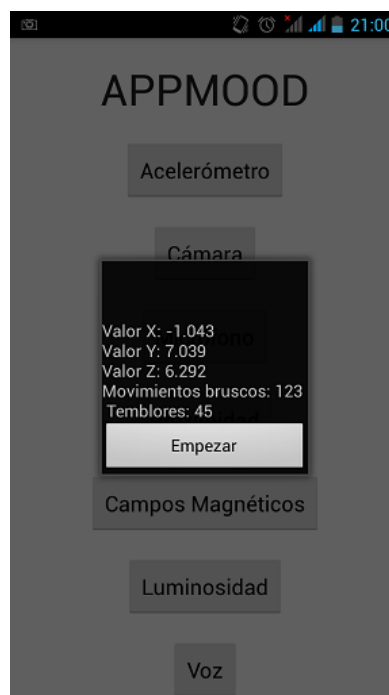


Figura 4.1. Resultados de la primera prueba del acelerómetro.

El resultado es el esperado, pues se han obtenido 123 movimientos bruscos, lo cual es aceptable, ya que si el dispositivo no se mueve se ha comprobado que no aumenta esa cifra.

Prueba nº2: comprobar que el acelerómetro recoge correctamente datos de temblores. Para ello se ha puesto en marcha dicho sensor y se ha fingido temblor en la mano.

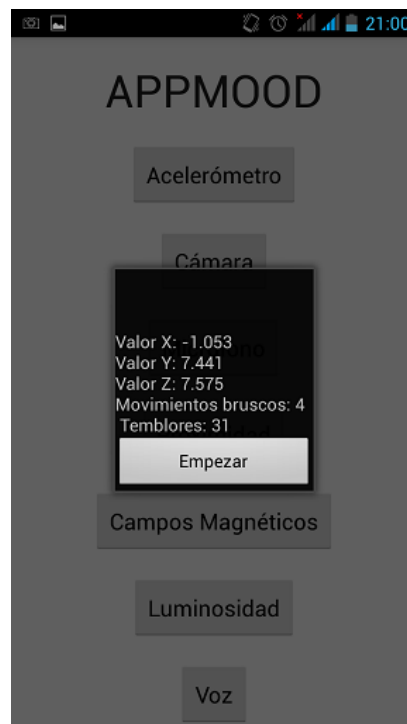


Figura 4.2. Resultados de la segunda prueba del acelerómetro.

El resultado es el esperado, pues se han obtenido 31 movimientos bruscos, lo cual es aceptable, ya que si el dispositivo no se mueve se ha comprobado que no aumenta esa cifra.

Prueba nº3: comprobar que la cámara puede detectar sonrisas. Para ello se ha puesto en marcha dicho sensor y se ha sonreído a la cámara.

El resultado ha sido el esperado puesto que se ha dibujado un rectángulo rosa alrededor de la sonrisa, además también se han detectado la cara y los ojos puesto que los ha rectangulado en verde y negro.

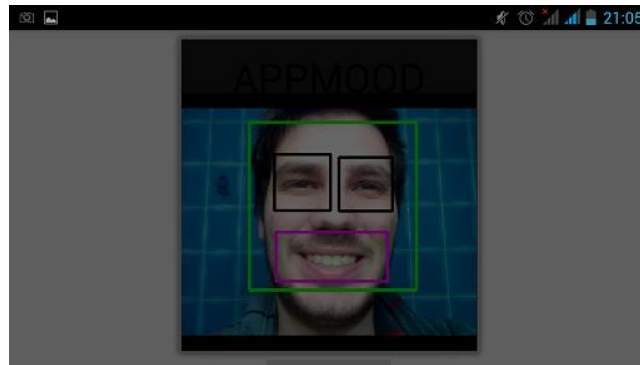


Figura 4.3. Resultados de la primera prueba de la cámara.

Prueba nº4: comprobar que la cámara puede detectar ojos cerrados. Para ello se ha puesto en marcha dicho sensor, se ha puesto la cámara enfocando la cara y se han cerrado los ojos.

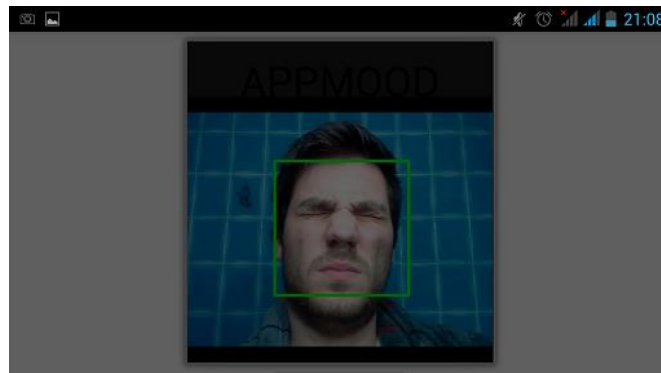


Figura 4.4. Resultados de la segunda prueba de la cámara.

El resultado ha sido el esperado puesto que no se han dibujado rectángulos negros alrededor de los ojos, mientras que la cara sí se ha recuadrado.

Cabe destacar que tanto en esta prueba como en la anterior, se ha necesitado de una buena iluminación.

Prueba nº5: comprobar que el sensor de proximidad detecta cambios de proximidad. Para ello se ha puesto en marcha dicho sensor, y se ha pasado la mano por delante del dispositivo a menos de 1cm.

El resultado ha sido el esperado puesto que se ha obtenido un valor de 5, que es exactamente las veces que se ha acercado la mano, no incrementándose este valor si no tiene nada cerca del sensor.



Figura 4.5. Resultados de la prueba al sensor de proximidad.

Prueba nº6: comprobar que el micrófono cuenta las palabras que pronuncia el usuario, así como el número de malsonantes.



Figura 4.6. Resultados de la prueba al micrófono.

El resultado ha sido el esperado pues ha indicado que se ha pronunciado una palabra malsonante, como así ha sido. En la base de datos se ha comprobado que ha contabilizado 4 palabras en total.

Prueba nº7: comprobar que el sensor de campos magnéticos detecta cambios en el campo magnético alrededor del usuario. Para ello se ha activado dicho sensor y se han comprobado los cambios registrados.



Figura 4.7. Resultados de la prueba al sensor de campos magnéticos.

Se podría decir que los resultados no son del todo satisfactorios, puesto que sí recoge bien los datos de los sensores pero no se puede considerar si los cambios magnéticos son debidos a la emoción del usuario. Acercando el dispositivo a otros aparatos electrónicos sí que aumenta claramente el número de cambios. Habría que estudiar más en profundidad si las emociones negativas afectan a este sensor, puesto que estas emociones no se pueden simular como sí se puede hacer en los demás sensores.

Prueba nº8: comprobar que el sensor de luminosidad detecta cambios en la luz ambiental. Para ello se ha activado dicho sensor y se ha sometido al dispositivo a cambios de luz.

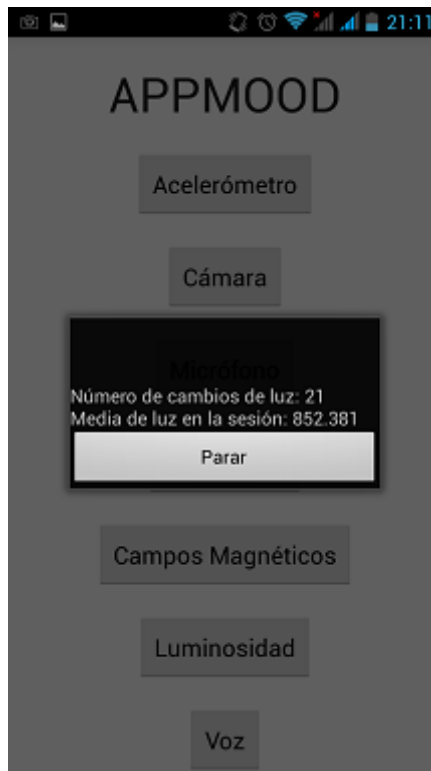


Figura 4.8. Resultados de la prueba al sensor de luminosidad.

Los resultados han sido los esperados, puesto que se ha movido el dispositivo acercándolo a la pantalla de un ordenador, quitándolo y apuntando a una bombilla y ha detectado todos los cambios, haciendo además su media correctamente.

Prueba nº9: comprobar que el medidor de decibelios registra los cambios irregulares en la voz y los gritos. Para ello se ha activado dicho sensor y se ha hablado y alzado la voz delante del dispositivo.

Los resultados han sido satisfactorios puesto que se han registrado 8 cambios al hablar con un tono irregular de voz. Además se han registrado dos gritos, uno al gritar y otro al dar una palmada.



Figura 4.9. Resultados de la prueba al medidor de decibelios.

4.2. Pruebas de la aplicación AppMood

Estas pruebas corresponden a las pruebas de sistema de la aplicación completa.

Prueba nº10: comprobar que la base de datos recoge correctamente los datos de cada sensor.

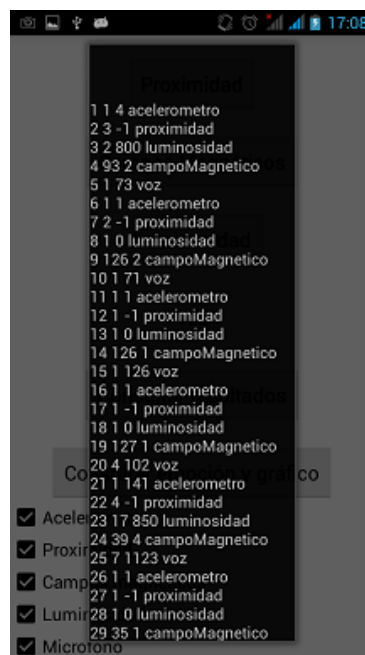


Figura 4.10. Resultados de la prueba a la base de datos.

Los resultados son satisfactorios puesto que se ha comprobado que se han insertado los nombres de los sensores junto con las señales emocionales medidas.

Prueba n°11: comprobar que la representación de la emoción es la adecuada. Para ello se fingirá una emoción y se comprobará si la sitúa bien en el gráfico. Esta prueba requiere que la aplicación tenga una base de datos con bastantes registros con los que comparar los últimos datos. Se fingirá una emoción del cuadrante inferior derecho, para ello se lanzará el servicio y se actuará positiva y relajadamente, como por ejemplo, que el acelerómetro no registre cambios y tampoco temblores.



Figura 4.11. Resultados de la prueba interpretando una emoción.

Los resultados son los adecuados puesto que ha situado la emoción en el cuadrante objetivo. No se puede determinar si es el punto más adecuado para la emoción, puesto que es fingida y la base de datos quizá no tenga demasiados datos con los que comparar, pero aun así es bastante aproximada. Adicionalmente se ha comprobado que fingiendo una emoción de los otros 3 cuadrantes también las sitúa correctamente.

Prueba n°12: comprobar que el sensor lanza una correcta notificación y canción. Para ello se activará el sensor y se fingirá la misma emoción que la de la prueba 11 durante un minuto.

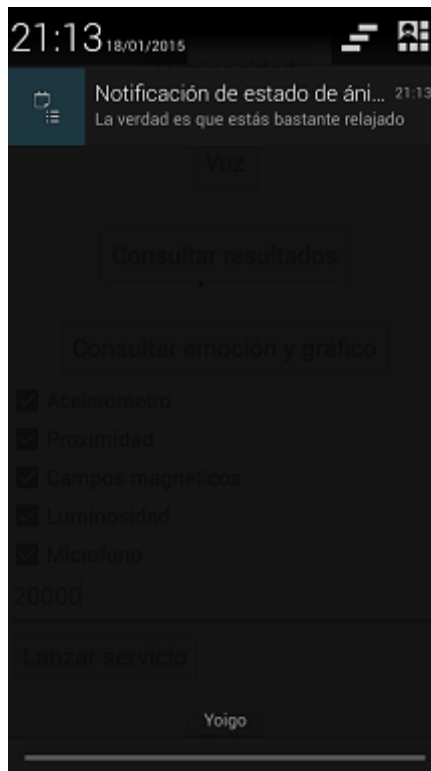


Figura 4.12. Resultados de la prueba del sensor interpretando una emoción.

Los resultados son los esperados puesto que lanza una notificación comentando lo relajado que está el usuario y reproduciendo una canción tranquila. Adicionalmente se ha comprobado el lanzamiento de notificaciones y canciones para los otros tres cuadrantes, comprobando también que son adecuadas.

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

5.1. Conclusiones

El objetivo principal de este trabajo era la demostración de las capacidades de los Smartphone como máquinas afectivas, concretamente para la inferencia de emociones a partir de los datos de los que puede disponer, y se puede concluir que ha sido cumplido de manera bastante satisfactoria. Probablemente los sensores que vienen de serie en los terminales comercializados no sean los suficientes como para determinar correctamente y de forma sencilla el estado emocional del usuario, pero mediante los datos que nos ofrecen podemos al menos atisbar el estado más probable en el que se encuentra el usuario, algo que se puede mejorar mediante sensores conectables al Smartphone o mediante vestibles.

El objetivo secundario de este trabajo era el uso que se le puede dar a esta información para mejorar la interacción con el usuario por parte de la aplicación, algo que se ha conseguido realizando una simple aplicación de recomendación de música respecto al estado emocional del usuario.

Por último, cabe destacar que se ha comprobado mediante la realización de las investigaciones para este trabajo que la computación afectiva es una potente perspectiva tecnológica para la mejora de los ordenadores y el software actual, ya que aporta valiosa información sobre el usuario, algo que ayudará al sistema a ofrecer una mejor experiencia y más personal experiencia al usuario.

5.2. Trabajo futuro

En el área de la computación afectiva queda mucho trabajo por realizar, debido a que pocas aplicaciones y sistemas actuales hacen uso de ella para todo lo que nos puede ofrecer.

Por una parte, la aplicación desarrollada en este trabajo se puede mejorar mediante la incorporación de más datos que nos puedan ofrecer sensores externos, así como mejorando el motor de inferencia o modificándolo por otro más sofisticado. Por ejemplo, con un sistema que utilice preguntas destinadas al usuario para aprender de

él, que tenga en cuenta los datos del sensor más adecuado dejando de lado los demás, o que a determinadas horas del día considere los sensores más adecuados.

Por otra parte, la de la computación afectiva en general, existe una cantidad ingente de ideas en las que poder aplicar esta tecnología, a continuación se enumeran algunas de ellas:

- Espejo afectivo: simular conversaciones con un ordenador, como en una video-llamada, para prepararse para entrevistas, discursos, etc. El ordenador avisa si el usuario cambia el tono de voz, si suena nervioso, si no mantiene contacto visual... El ordenador también puede interpretar emociones para hacerlo más real.
- Emails con emociones: los correos están formados por texto plano y a veces es difícil interpretar cómo se quiere expresar. El ordenador los puede leer con un tono determinado o cambiar la interfaz.
- Ayuda a autistas: indicar el estado de ánimo del interlocutor para que el usuario con dificultades pueda ir aprendiendo lo que quieren decir los demás.
- Videojuegos: se adaptan al jugador. Se pone más fácil si el usuario está estresado, recompensa si se relaja...
- Autoaprendizaje: adaptar un software a la hora de enseñar algo respecto a las emociones (curiosidad, estrés...).
- -Reproducir música: previamente elegida para cada estado de ánimo al final del día, o recomendarla.
- Mejora de tráileres de cine: hacerlos más interesantes con las partes que más hayan gustado, habiendo recopilado las reacciones de usuarios que ya los han visto.
- Agentes que aprenden de las preferencias: el ordenador se da cuenta de si no le gusta una determinada noticia al usuario y no volverá a mostrar de ese tipo la próxima vez. También recomienda ropa o música si observa que le gustan.
- Un tamagotchi o mascota virtual que reconozca emociones como una mascota real.

REFERENCIAS

- [1] Rosalind W. Picard, "*Affective Computing*"
- [2] Paul Ekman, "*Emotion in the Human Face*"
- [3] Paul Ekman, "*Unmasking the face*"
- [4] R. Cowie, "*Emotion Recognition in Human-Computer*"
- [5] Irina Gorodnitsky, "*Pitch modulation in emotional speech by non-emotive factors*"
- [6] James A. Russell, "*A circumplex model of affect*"
- [7] J. Andrew Armour, "*Neurocardiology: Anatomical and Functional Principles*"
- [8] Kiran K. Rachuri, "*EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research*"
- [9] Paul Ekman, "*An argument for basic emotions*"
- [10] W.M. Wundt, "*Outlines of Psychology*"
- [11] <http://www.abc.es/tecnologia/moviles-telefonía/20140911/abci-penetracion-smartphone-espana-201409111850.html/> , Diciembre 2014
- [12] <http://www.affectiva.com/>, Diciembre 2014
- [13] <https://www.empatica.com/>, Diciembre 2014
- [14] http://developer.android.com/guide/topics/sensors/sensors_overview.html/, Diciembre 2014
- [15] <http://www.gartner.com/newsroom/id/2623415/> Diciembre 2014
- [16] <http://www.electronicaestudio.com/sensores.htm/> Diciembre 2014
- [17] <http://www.abc.es/tecnologia/moviles-telefonía/20140715/abci-xiaomi-smartphone-vendidos-201407151608.html/> Diciembre 2014
- [18] play.google.com/store/apps/details?id=imoblife.androidsensorbox&hl=es/ Diciembre 2014
- [19] <http://es.blastingnews.com/tecnologia/2014/12/que-porcentaje-ocupan-los-moviles-chinos-en-el-mundo-00193175.html/> Diciembre 2014
- [20] http://en.wikipedia.org/wiki/Emotion_classification/ Diciembre 2014
- [21] <http://rufadas.com/wp-content/uploads/2008/02/malsonantes.pdf/> Noviembre 2014
- [22] <http://opencv.org/> Octubre 2014
- [23] <http://www.iis.fraunhofer.de/en/ff/bsy/tech/bildanalyse/shore-gesichtsdetektion.html> Noviembre 2014

[24] <http://www.nviso.ch/> Noviembre 2014

[25] <http://emospeech.net/> Noviembre 2014